

**பகுதி - I தமிழ்ப் பாடத்திட்டம் (2018 - 2019)**  
**முதல்பருவம்**  
**(இளநிலை அறிவியல் பட்ட வகுப்புகளுக்குரியது)**  
**(For I-UG Science Degree Classes) 18LSU101**

**பாடத்திட்டப் பொதுநோக்கம்**

- கற்றல் வழி சிந்தனைத் திறனையும், கருத்து வெளிப்பாட்டுத் திறனையும், மேம்படுத்துதல்.
- ஆய்வுநோக்கை மேம்படுத்துதல்.
- இலக்கியங்கள் உணர்த்தும் வாழ்வின் நுட்பமான பகுதிகளை உணர்த்துதல்.
- மனித மனத்தினைப் பக்குவப்படுத்துதலில் இலக்கியம் தரும் பங்கினை உணர்த்துதல்.
- வளர்ந்து வரும் சமூகத்தில் அறஉணர்வு, பண்பாடு போன்றவை குறித்து அறிவூட்டல்.
- அரசுத் தேர்வுகளுக்கு மாணவர்களை ஆயத்தமாக்குதல்.

**பாடத்திட்டப் பயன் விளைவு**

- இந்திய குடியுரிமைப் பணி முதலான போட்டித் தேர்வுகளில், விருப்பப் பாடமாக இடம்பெறுகின்ற, 'தமிழ் இலக்கிய வரலாறு' குறித்த முழுமையான அறிமுகம் பெற்றிருத்தல்.
- கல்வெட்டியல், ஓலைச்சுவடியியல் மற்றும் தொல்லியல் சார்ந்த ஆவணத் தேடலுக்குரிய ஆய்வுமனப்பான்மையுடன், இலக்கியங்களை அணுகுதல்.
- தமிழின் வளர்ச்சித் துறையாகிய, 'அறிவியல் தமிழ்' ; 'இணைய தமிழ்' குறித்த பன்னோக்கு அணுகுமுறையிலான ஆய்வுச் சிந்தனை மேம்பாடு.
- வேலைவாய்ப்புக்குரிய சுயதிறன் மேம்பாட்டுடன், படைப்பாக்கத்திறன் மேம்பாடும் பெற்றிருத்தல் .
- சமுதாய மற்றும் வாழ்வியல் மதிப்புகளைப் பேணுவதற்குக் கருவியாக இலக்கியங்களை நாடுகின்ற மனப்பான்மை வளர்ச்சி.
- மொழிபெயப்புத் துறைசார்ந்த வேலைவாய்ப்புத் திறன் பெற்றிருத்தல்

பகுதி- I, தமிழ்                      முதல் பருவம்                      18LSU101 :  
 தமிழ் முதல் தாள்                      4-H,4-C  
 (இளநிலை அறிவியல் பட்ட வகுப்புகளுக்குரியது)  
 (For I-UG Science Degree Classes)

**அலகு - I : இக்கால இலக்கியம்:                      (10 மணிநேரம்)**

கல்வி : மகாகவி பாரதியார் - சுயசரிதை - ஆங்கிலக் கல்வி.

இன்றைய நிலை : கவிமணி தேசிக விநாயகம்பிள்ளை-ஒற்றுமையே உயிர்நிலை.

மனிதநேயம் : கவிஞர் சிற்பி பாலசுப்பிரமணியன் -மலையாளக் காற்று.

சூழலியல் : கவிஞர் வைதீஸ்வரன் - விரல் மீட்டிய மழை.

பெண்ணியம் : கவிஞர் சுகந்தி சுப்பிரமணியம் - புதையுண்ட வாழ்க்கை.

**அலகு - II : அற இலக்கியம்:                      (8 மணிநேரம்)**

கொன்றை வேந்தன்: 1-50 பாடல்கள்

திருக்குறள்: பண்புடைமை, வினைத்திட்டம் - 20 குறள்கள்

பழமொழி நானூறு: 5 பாடல்கள்

**அலகு - III : சிற்றிலக்கியம்:                      (8 மணிநேரம்)**

மூவருலா: 1-26 கண்ணிகள்

திருச்செந்தூர் முருகன் பிள்ளைத்தமிழ்: 2 பாடல்கள்

கலிங்கத்துப் பரணி: போர்பாடியது - 9 பாடல்கள்

**அலகு - IV : கட்டுரை:                      (8 மணிநேரம்)**

1. உயர்தனிச் செம்மொழி - பரிதிமாற்கலைஞர்
2. கட்டிடக்கலை - அ. இராசமாணிக்கனார்
3. வாழ்க்கை - இளவழகனார்
4. ஆளுமைத்திறன் அறிவோம் - ஸ்ரீகண்ணன்
5. மணற்கேணி - நெ.து.சுந்தரவடிவேலு

**அலகு- V : மொழிப்பயிற்சி:****(6 மணிநேரம்)**

1. பொருத்தமான தமிழ்ச் சொற்களைப் பயன்படுத்துதல்
2. செய்யுள் பொருளுணர் திறன்
3. மொழிபெயர்ப்புப் பயிற்சிகள்
4. கடிதங்கள் மற்றும் விண்ணப்பங்கள் எழுதுதல்

**பாட நூல்: கற்பகச்சோலை – தமிழ் ஏடு.****கற்பகம் உயர்கல்வி கலைக்கழகத் தமிழ்த்துறை வெளியீடு.**

---

---

**Semester – I**

**18ITU101    PROGRAMMING FUNDAMENTALS USING C / C++    4H – 4C**

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To impart adequate knowledge on the need of programming languages and problem solving techniques.
- To develop programming skills using the fundamentals and basics of C Language.
- To enable effective usage of arrays, structures, functions, pointers and to implement the memory management concepts.
- To teach the issues in file organization and the usage of file systems.
- To learn the characteristics of an object-oriented programming language: data abstraction and information hiding, inheritance, and dynamic binding of the messages to the methods.
- To learn how to write inline functions for efficiency and performance.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Obtain the knowledge about the number systems this will be very useful for bitwise operations.
2. Develop programs using the basic elements like control statements, Arrays and Strings.
3. Understand about the dynamic memory allocation using pointers which is essential for utilizing memory
4. Understand about the code reusability with the help of user defined functions.
5. Develop advanced applications using enumerated data types, function pointers and nested structures, the basic object-oriented design principles in computer problem solving.
6. Learn the basics of file handling mechanism that is essential for understanding the concepts in database management systems, the uses of preprocessors and various header file directives, the characteristics of an object-oriented programming language in a program.

**Unit I - INTRODUCTION TO C AND C++**

History of C and C++, Overview of Procedural Programming and Object-Orientation Programming, Using main() function, Compiling and Executing Simple Programs in C++.

**Data Types, Variables, Constants, Operators and Basic I/O:**

Declaring, Defining and Initializing Variables, Course Objectives of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putchar), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.h etc).

**Expressions, Conditional Statements and Iterative Statements:**

Simple Expressions in C++ (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch-case construct), Understanding syntax and utility of Iterative Statements

(while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)

## **Unit II - FUNCTIONS AND ARRAYS**

Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments.

Creating and Using One Dimensional Arrays ( Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays.

## **Unit III - DERIVED DATA TYPES (STRUCTURES AND UNIONS)**

Understanding utility of structures and unions, Declaring, initializing and using simple structures and unions, Manipulating individual members of structures and unions, Array of Structures, Individual data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.

### **Pointers and References in C++:**

Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems with Pointers, Passing pointers as function arguments, Returning a pointer from a function, using arrays as pointers, Passing arrays to functions. Pointers vs. References, Declaring and initializing references, using references as function arguments and function return values

## **Unit IV - MEMORY ALLOCATION IN C++**

Differentiating between static and dynamic memory allocation, use of malloc, calloc and free functions, use of new and delete operators, storage of variables in static and dynamic memory allocation.

### **File I/O, Preprocessor Directives:**

Opening and closing a file (use of fstream header file, ifstream, ofstream and fstream classes), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros.

## **Unit V - USING CLASSES IN C++**

Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Function overloading in classes, Class Variables & Functions, Objects as parameters, Specifying the Protected and Private Access, Copy Constructors, Overview of Template classes and their use.

### **Overview of Function Overloading and Operator Overloading:**

Need of Overloading functions and operators, Overloading functions by number and type of arguments, Looking at an operator as a function call, Overloading Operators (including assignment operators, unary operators).

### **Inheritance, Polymorphism and Exception Handling:**

Introduction to Inheritance (Multi-Level Inheritance, Multiple Inheritance), Polymorphism (Virtual Functions, Pure Virtual Functions), Basics Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions, Restricting exceptions, Rethrowing exceptions.

### Suggested Readings

1. Herbtz Schildt. (2003). C++: The Complete Reference (4th ed.) McGraw Hill, New Delhi.
2. Bjarne Stroustrup. (2013). The C++ Programming Language(4th ed.). Addison-Wesley, New Delhi.
3. Bjarne Stroustrup. (2014). Programming, Principles and Practice using C++(2<sup>nd</sup> ed)Addison-Wesley, New Delhi.
4. Balaguruswamy, E. (2008). Object Oriented Programming with C++. Tata McGraw-Hill Education, New Delhi.
5. Paul Deitel., & Harvey Deitel. (2011). C++ How to Program (8th ed.). Prentice Hall, New Delhi.
6. John, R. Hubbard. (2000). Programming with C++- (2nd ed.). Schaum's Series.
7. Andrew Koeni., Barbara, E. Moo. (2000). Accelerated C++. Addison-Wesley.
8. Scott Meyers. (2005). Effective C++ (3rd ed.).Addison-Wesley,.
9. Harry, H. Chaudhary. (2014). Head First C++ Programming: The Definitive Beginner's Guide. LLC USA: First Create space Inc, O-D Publishing.
10. Walter Savitch.( 2007) Problem Solving with C++, Pearson Education,.
11. Stanley, B. Lippman., Josee Lajoie., & Barbara, E. Moo. (2012). C++ Primer, 5th ed.). Addison-Wesley

### Web Sites

1. <http://www.cs.cf.ac.uk/Dave/C/CE.html>
2. <http://www2.its.strath.ac.uk/courses/c/>
3. <http://www.iu.hio.no/~mark/CTutorial/CTutorial.html>
4. <http://www.cplusplus.com/doc/tutorial/>
5. [www.cplusplus.com/](http://www.cplusplus.com/)
6. [www.cppreference.com/](http://www.cppreference.com/)

---

<b>18ITU102</b>	<b>COMPUTER SYSTEM ARCHITECTURE</b>	<b>Semester – I</b> <b>4H – 4C</b>
-----------------	-------------------------------------	---------------------------------------

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To enable the students to gain knowledge on the architecture of modern computer.
- To understand how computer stores positive and negative numbers and to perform arithmetic operation of positive and negative numbers.
- To learn about logic gates and solve problems using Boolean algebra.
- To understand the simplification of circuits like adders, subtractors, multiplexers, encoders.
- To understand the basic computer organization and design.
- To learn Cache memory and its importance

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Acquire a basic knowledge about computer system architecture, digital circuits and the low - level programming skills.
2. Understand the inner workings and performance capabilities of advanced microprocessors.
3. Solve the problems using Boolean algebra
4. Understand the basic computer organization and design.
5. Learn about Cache memory and its importance
6. Solve the binary arithmetic problems and conversion among the number systems

**Unit I - INTRODUCTION**

Logic gates, Boolean algebra, circuit simplification, combinational circuits: Adders and Subtractors –Multiplexers and De multiplexers – Encoders and Decoders- sequential circuits: Flip Flop's, registers, counters and memory Units.

**Unit II - DATA REPRESENTATION AND BASIC COMPUTER ARITHMETIC**

Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison, multiplication and division algorithms for integers

**Unit III - BASIC COMPUTER ORGANIZATION AND DESIGN**

Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt, Interconnection Structures, Bus Interconnection design of basic computer.

**Unit IV - CENTRAL PROCESSING Unit**

Register organization, arithmetic and logical micro-operations, stack organization, micro

programmed control. Instruction formats, addressing modes, instruction codes, machine language, assembly language, input output programming, RISC, CISC architectures, pipelining and parallel architecture.

### **Unit V - MEMORY AND INPUT-OUTPUT ORGANIZATION**

Cache memory, Associative memory, mapping Input / Output: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels.

### **Suggested Readings**

- 1.M.Mano. (1992). Computer System Architecture. Pearson Education.
2. Dos Reis, A. J. (2009). Assembly Language and Computer Architecture using C++ and JAVA. Course Technology
3. Stallings, W. (2010). Computer Organization and Architecture Designing for Performance (8th ed.) Prentice Hall of India, New Delhi.
4. Mano, M.M. (2013). Digital Design. Pearson Education Asia, New Delhi.
5. Carl Hamacher. (2012). Computer Organization (5th ed.). McGrawHill, New Delhi.



<b>18ITU103</b>	<b>COMPUTER FUNDAMENTALS</b>	<b>Semester – I</b> <b>4H – 4C</b>
<b>Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100</b> <b>End Semester Exam : 3 Hours</b>		

### Course Objectives (CO)

- To identify types of computers, how they process information and how individual computers interact with other computing systems and devices.
- To identify the function of computer hardware components.
- To identify the factors that goes into an individual or organizational decision on how to purchase computer equipment.
- To identify how to maintain computer equipment and solve common problems relating to computer hardware.
- To identify how software and hardware work together to perform computing tasks and how software is developed and upgraded.
- To identify different types of software, general concepts relating to software categories, and the tasks to which each type of software is most suited or not suited.

### Course Outcomes (COs)

After the completion of this course, a successful student will be able to:

1. Understand the meaning and basic components of a computer system,
2. Define and distinguish Hardware and Software components of computer system,
3. Explain and identify different computing machines during the evolution of computer system, gain knowledge about five generations of computer system,
4. Identify and discuss the functional Units of a computer system, identify the various input and output Units and explain their purposes
5. Understand the role of CPU and its components, understand the concept and need of primary and secondary memory, discuss the advantages, limitations and applications of computers.
6. Understand the classification of computers, distinguish the computers on the basis of purpose, technology and size

### Unit I – INTRODUCTION

Introduction to computer system, uses, types. **Data Representation:** Number systems and character representation, binary arithmetic. **Human Computer Interface:** Types of software, Operating system as user interface, utility programs.

### Unit II – DEVICES

Input and output devices (with connections and practical demo), keyboard, mouse, joystick, scanner, OCR, OMR, bar code reader, web camera, monitor, printer, plotter.

**Unit III – MEMORY**

Primary, secondary, auxiliary memory, RAM, ROM, cache memory, hard disks, optical disks.

**Unit IV - COMPUTER ORGANISATION AND ARCHITECTURE**

C.P.U., registers, system bus, main memory Unit, cache memory, Inside a computer, SMPS, Motherboard, Ports and Interfaces, expansion cards, ribbon cables, memory chips, processors.

**Unit V - OVERVIEW OF EMERGING TECHNOLOGIES**

Bluetooth, cloud computing, big data, data mining, mobile computing and embedded systems.

**Suggested Readings**

1. Goel, A. (2010). Computer Fundamentals. Pearson Education, New Delhi.
2. Aksoy, P., & DeNardis, L. (2006). Introduction to Information Technology. Cengage Learning, New Delhi.
3. Sinha, P. K., & Sinha, P. (2007). Fundamentals of Computers. BPB Publishers, New Delhi.

---

<b>18ITU111</b>	<b>PROGRAMMING FUNDAMENTALS USING C / C++ - PRACTICAL</b>	<b>Semester – I 4H – 2C</b>
-----------------	---	---------------------------------

---

**Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To impart adequate knowledge on the need of programming languages and problem solving techniques.
- To develop programming skills using the fundamentals and basics of C Language.
- To enable effective usage of arrays, structures, functions, pointers and to implement the memory management concepts.
- To teach the issues in file organization and the usage of file systems.
- To learn the characteristics of an object-oriented programming language: data abstraction and information hiding, inheritance, and dynamic binding of the messages to the methods.
- To learn how to write inline functions for efficiency and performance.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Obtain the knowledge about the number systems this will be very useful for bitwise operations.
2. Develop programs using the basic elements like control statements, Arrays and Strings.
3. Understand about the dynamic memory allocation using pointers which is essential for utilizing memory
4. Understand about the code reusability with the help of user defined functions.
5. Develop advanced applications using enumerated data types, function pointers and nested structures, the basic object-oriented design principles in computer problem solving.
6. Learn the basics of file handling mechanism that is essential for understanding the concepts in database management systems, the uses of preprocessors and various header file directives, the characteristics of an object-oriented programming language in a program.

**List of Programs**

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first n terms of the following series  $S = 1 + 1/2 + 1/3 + 1/4 + \dots$
4. WAP to compute the sum of the first n terms of the following series  $S = 1 - 2 + 3 - 4 + 5 - \dots$

5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
- 7.WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. WAP to print a triangle of stars as follows (take number of lines from user):

```
*  
***  
*****  
*****  
*****
```

10. WAP to perform following actions on an array entered by the user:
  - i) Print the even-valued elements
  - ii) Print the odd-valued elements
  - iii) Calculate and print the sum and average of the elements of array
  - iv) Print the maximum and minimum element of array
  - v) Remove the duplicates from the array
  - vi) Print the array in reverse order

The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.

11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.

16. Write a menu driven program to perform following operations on strings:
- a) Show address of each character in string
  - b) Concatenate two strings without using strcat function.
  - c) Concatenate two strings using strcat function.
  - d) Compare two strings
  - e) Calculate length of the string (use pointers)
  - f) Convert all lowercase characters to uppercase
  - g) Convert all uppercase characters to lowercase
  - h) Calculate number of vowels
- i) Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i)using recursion, (ii) using iteration
19. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation):
- a) Sum b) Difference c) Product d) Transpose
22. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
23. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.
24. Create a class Box containing length, breath and height. Include following methods in it:
- a) Calculate surface Area
  - b) Calculate Volume
  - c) Increment, Overload ++ operator (both prefix & postfix)
  - d) Decrement, Overload -- operator (both prefix & postfix)
  - e) Overload operator == (to check equality of two boxes), as a friend function

f) Overload Assignment operator

g) Check if it is a Cube or cuboid

Write a program which takes input from the user for length, breath and height to test the above class.

25. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.

26. Write a program to retrieve the student information from file created in previous question and print it in following format:

Roll No. Name Marks

27. Copy the contents of one text file to another file, after removing all whitespaces.

28. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.

29. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

### Suggested Readings

1. Bjarne Stroustrup. (2013). The C++ Programming Language(4th ed.). Addison-Wesley, New Delhi.
2. Bjarne Stroustrup. (2014). Programming, Principles and Practice using C++(2<sup>nd</sup> edAddison-Wesley, New Delhi.
3. Balaguruswamy, E. (2008). Object Oriented Programming with C++. Tata McGraw-Hill Education, New Delhi.
4. Paul Deitel., & Harvey Deitel. (2011). C++ How to Program (8th ed.). Prentice Hall, New Delhi.
5. Harry, H. Chaudhary. (2014). Head First C++ Programming: The Definitive Beginner's Guide. LLC USA: First Create space Inc, O-D Publishing.

### Web Sites

1. <http://www2.its.strath.ac.uk/courses/c/>
2. <http://www.iu.hio.no/~mark/CTutorial/CTutorial.html>
3. <http://www.cplusplus.com/doc/tutorial/>
4. [www.cplusplus.com/](http://www.cplusplus.com/)
5. [www.cppreference.com/](http://www.cppreference.com/)

**18ITU112      COMPUTER SYSTEM ARCHITECTURE  
- PRACTICAL****Semester – I  
3H – 2C****Instruction Hours / week: L: 0 T: 0 P: 3      Marks: Internal : 40    External : 60    Total: 100  
End Semester Exam : 3 Hours****Course Objectives (CO)**

- To enable the students to gain knowledge on the architecture of modern computer.
- To understand how computer stores positive and negative numbers and to perform arithmetic operation of positive and negative numbers.
- To learn about logic gates and solve problems using Boolean algebra.
- To understand the simplification of circuits like adders, subtractors, multiplexers, encoders.
- To understand the basic computer organization and design.
- To learn Cache memory and its importance

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Acquire a basic knowledge about computer system architecture, digital circuits and the low - level programming skills.
2. Understand the inner workings and performance capabilities of advanced microprocessors.
3. Solve the problems using Boolean algebra
4. Understand the basic computer organization and design.
5. Learn about Cache memory and its importance
6. Solve the binary arithmetic problems and conversion among the number systems

**List of Experiments**

(Any 8 Experiments)

1. Verification of Logic Gates
2. Code converters
3. Realization of Multiplexer using basic gates
4. Encoder and Decoder
5. Realization Half and Full adders
6. Realization of Subtractor
7. Realization of Parity generator
8. Flip-Flop Circuits
9. Digital to analog Converters
10. Demonstrate a Basic Arithmetic Computing operations

**Suggested Readings**

1.M.Mano. (1992). Computer System Architecture. Pearson Education.

2. Dos Reis, A. J. (2009). Assembly Language and Computer Architecture using C++ and JAVA. Course Technology
3. Stallings, W. (2010). Computer Organization and Architecture Designing for Performance (8th ed.) Prentice Hall of India, New Delhi.
4. Mano, M.M. (2013). Digital Design. Pearson Education Asia, New Delhi.
5. Carl Hamacher. (2012). Computer Organization (5th ed.). McGrawHill, New Delhi.



---

<b>18ITU113</b>	<b>COMPUTER FUNDAMENTALS - PRACTICAL</b>	<b>Semester – I</b> <b>3H – 2C</b>
-----------------	--	---------------------------------------

---

**Instruction Hours / week: L: 0 T: 0 P: 3    Marks: Internal : 40    External : 60 Total: 100**  
**End Semester Exam : 3 Hours**

### **Course Objectives (CO)**

- To create a document in Microsoft Word with formatting options, edit, save, and print documents to include documents with lists and tables, Format text and to use styles, add a header and footer to a document, add a graphic to a document.
- To write functions in Microsoft Excel to perform basic calculations and to convert number to text and text to number.
- To indicate the names and functions of the Excel interface components.
- To enter and edit data, Format data and cells.
- Construct formulas, including the use of built-in functions, and relative and absolute references.
- Create and modify charts.

### **Course Outcomes (COs)**

After completion of this course, the students will be able to

1. Modify text using various formatting options from the editing tools under the Home tab, set up section breaks to create different headers and footers for the odd and even pages within the document sections.
2. Demonstrate the mechanics and uses of Word tables to organize and present data, demonstrate working knowledge of using Word's themes and clip art to create a variety of visual effects.
3. Demonstrate working knowledge of Word's advanced formatting techniques and presentation styles,
4. Demonstrate applicable knowledge and uses of accepted business style formatting conventions.
5. Create and design a spreadsheet for general office use, demonstrate the basic mechanics and navigation of an Excel spreadsheet.
6. Demonstrate formatting techniques and presentation styles, demonstrate the use of basic functions and formulas

Practical exercises based on MS Office/ Open Office tools using document preparation and spreadsheet handling packages.

### MS Word

1. Prepare a **grocery list** having four columns (Serial number, The name of the product, quantity and price) for the month of April, 06.

- Font specifications for Title (Grocery List): 14-point Arial font in bold and italics.
- The headings of the columns should be in 12-point and bold.
- The rest of the document should be in 10-point Times New Roman.
- Leave a gap of 12-points after the title.

2. Create a **telephone directory**.

- The heading should be 16-point Arial Font in bold
- The rest of the document should use 10-point font size
- Other headings should use 10-point Courier New Font.
- The footer should show the page number as well as the date last updated.

3. Design a **time-table form** for your college.

- The first line should mention the name of the college in 16-point Arial Font and should be bold.
- The second line should give the course name/teacher's name and the department in 14-point Arial.
- Leave a gap of 12-points.
- The rest of the document should use 10-point Times New Roman font.
- The footer should contain your specifications as the designer and date of creation.

4. BPB Publications plans to release a new book designed as per your syllabus. Design the **first page of the book** as per the given specifications.

- The title of the book should appear in bold using 20-point Arial font.
- The name of the author and his qualifications should be in the center of the page in 16-point Arial font.
- At the bottom of the document should be the name of the publisher and address in 16-point Times New Roman.
- The details of the offices of the publisher (only location) should appear in the footer.

5. Create the following one page documents.

- a. Compose a note inviting friends to a get-together at your house, Including a list of things to bring with them.
- b. Design a certificate in landscape orientation with a border around the document.
- c. Design a Garage Sale sign.
- d. Make a sign outlining your rules for your bedroom at home, using a numbered list.

6. Create the following documents:

- (a) A newsletter with a headline and 2 columns in portrait orientation, including at least one image surrounded by text.

(b) Use a newsletter format to promote upcoming projects or events in your classroom or college.

7. Convert following text to a table, using comma as delimiter

Type the following as shown (do not bold).

**Color, Style, Item**

**Blue, A980, Van**

**Red, X023, Car**

**Green, YL724, Truck**

**Name, Age, Sex**

**Bob, 23, M**

**Linda, 46, F**

**Tom, 29, M**

8. Enter the following data into a table given on the next page.

Salesperson	Dolls	Trucks	Puzzles
Kennedy, Sally	1327	1423	1193
White, Pete	1421	3863	2934
Pillar, James	5214	3247	5467
York, George	2190	1278	1928
Banks, Jennifer	1201	2528	1203
Atwater, Kelly	4098	3079	2067
Pillar, James	5214	3247	5467
York, George	2190	1278	1928
Banks, Jennifer	1201	2528	1203
Atwater, Kelly	4098	3079	2067

Add a column Region (values: S, N, N,S,S,S) between the Salesperson and Dolls columns to the given table Sort your table data by Region and within Region by Salesperson in ascending order: In this exercise, you will add a new row to your table, place the word Total at the bottom of the Salesperson column, and sum the Dolls, Trucks, and Puzzles columns.

9. Wrapping of text around the image.

10. Following features of menu option must be covered

FILE Complete menu

EDIT Complete menu

VIEW Complete menu

INSERT Complete menu

FORMAT Complete menu

TABLE Complete menu

WINDOW Complete menu

HELP Complete menu

TOOLS All options except Online collaboration, Tools on Macro, Templates

**MS Excel**

1. Enter the Following data in Excel Sheet

<b>REGIONAL SALES PROJECTION</b>						
<b>State</b>	<b>Qtr1</b>	<b>Qtr2</b>	<b>Qtr3</b>	<b>QTR4</b>	<b>Qtr Total</b>	<b>Rate Amount</b>
Delhi	2020	2400	2100	3000	15	
Punjab	1100	1300	1500	1400	20	
U.P.	3000	3200	2600	2800	17	
Haryana	1800	2000	2200	2700	15	
Rajasthan	2100	2000	1800	2200	20	

**TOTAL  
AVERAGE**

(a) Apply Formatting as follow:

- i. Title in TIMES NEW ROMAN
- ii. Font Size - 14
- iii. Remaining text - ARIAL, Font Size -10
- iv. State names and Qtr. Heading Bold, Italic with Gray Fill Color.
- v. Numbers in two decimal places.
- vi. Qtr. Heading in center Alignment.
- vii. Apply Border to whole data.

(b) Calculate State and Qtr. Total

(c) Calculate Average for each quarter

(d) Calculate Amount = Rate \* Total .

2. Given the following worksheet

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
1	Roll No.	Name	Marks	Grade
2	1001	Sachin	99	
3	1002	Sehwag	65	

4	1003	Rahul	41	
5	1004	Sourav	89	
6	1005	Har Bhajan	56	

Calculate the grade of these students on the basis of following guidelines:

If Marks	Then Grade
$\geq 80$	A+
$\geq 60 < 80$	A
$\geq 50 < 60$	B
$< 50$	F

3. Given the following worksheet

	A	B	C	D	E	F	
1	Salesman			Sales in (Rs.)			
2	No.	Qtr1	Qtr2	Qtr3	Qtr4	Total	Commission
3	S001	5000	8500	12000	9000		
4	S002	7000	4000	7500	11000		
5	S003	4000	9000	6500	8200		
6	S004	5500	6900	4500	10500		
7	S005	7400	8500	9200	8300		
8	S006	5300	7600	9800	6100		

Calculate the commission earned by the salesmen on the basis of following Candidates:

If Total Sales	Commission
$< 20000$	0% of sales
$> 20000$ and $< 25000$	4% of sales
$> 25000$ and $< 30000$	5.5% of sales
$> 30000$ and $< 35000$	8% of sales
$\geq 35000$	11% of sales

The total sales is sum of sales of all the four quarters.

4. A company XYZ Ltd. pays a monthly salary to its employees which consists of basic salary, allowances & deductions. The details of allowances and deductions are as follows:

#### Allowances

- HRA Dependent on Basic
  - 30% of Basic if Basic  $\leq 1000$
  - 25% of Basic if Basic  $> 1000$  & Basic  $\leq 3000$
  - 20% of Basic if Basic  $> 3000$
- DA Fixed for all employees, 30% of Basic
- Conveyance Allowance Rs. 50/- if Basic is  $\leq 1000$

Rs. 75/- if Basic >1000 & Basic ≤ 2000

Rs. 100 if Basic >2000

- Entertainment Allowance NIL if Basic is ≤ 1000 Rs. 100/- if Basic > 1000

### Deductions

- Provident Fund 6% of Basic
- Group Insurance Premium Rs. 40/- if Basic is ≤ 1500  
Rs. 60/- if Basic > 1500 & Basic ≤ 3000  
Rs. 80/- if Basic > 3000

Calculate the following:

Gross Salary = Basic + HRA + DA + Conveyance + Entertainment

Total deduction = Provident Fund + Group Insurance Premium

Net Salary = Gross Salary – Total Deduction.

5. Create Payment Table for a fixed Principal amount, variable rate of interests and time in the format below:

No. of Installments	5%	6%	7%	8%	9%
3	XX	XX	XX	XX	XX
4	XX	XX	XX	XX	XX
5	XX	XX	XX	XX	XX
6	XX	XX	XX	XX	XX

6. Use an array formula to calculate Simple Interest for given principal amounts given the rate of Interest and time

Rate of Interest	8%
Time	5 Years
Principal	Simple Interest
1000	?
18000	?
5200	?

7. The following table gives year wise sale figure of five salesmen in Rs.

Salesman	2000	2001	2002	2003
S1	10000	12000	20000	50000
S2	15000	18000	50000	60000
S3	20000	22000	70000	70000
S4	30000	30000	100000	80000
S5	40000	45000	125000	90000

(a) Calculate total sale year wise.

(b) Calculate the net sale made by each salesman

- (c) Calculate the maximum sale made by the salesman  
 (d) Calculate the commission for each salesman under the condition.  
     (i) If total sales >4,00,000 give 5% commission on total sale made by the salesman.  
     (ii) Otherwise give 2% commission.  
 (e) Draw a bar graph representing the sale made by each salesman.  
 (f) Draw a pie graph representing the sale made by salesman in 2000.

8. Enter the following data in Excel Sheet

### PERSONAL BUDGET FOR FIRST QUARTER

**Monthly Income (Net): 1,475**

EXPENSES	JAN	FEB	MARCH	QUARTER TOTAL	QUARTER AVERAGE
Rent	600.00	600.00	600.00		
Telephone	48.25	43.50	60.00		
Utilities	67.27	110.00	70.00		
Credit Card	200.00	110.00	70.00		
Oil	100.00	150.00	90.00		
AV to					
Insurance	150.00				
Cable TV	40.75	40.75	40.75		

#### Monthly Total

Calculate Quarter total and Quarter average.

- (a) Calculate Monthly total.  
 (b) Surplus = Monthly income - Monthly total.  
 (c) What would be total surplus if monthly income is 1500.  
 (d) How much does telephone expense for March differ from quarter average.  
 (e) Create a 3D column graph for telephone and utilities. (f) Create a pie chart for monthly expenses.

9. Enter the following data in Excel Sheet

### TOTAL REVENUE EARNED FOR SAM'S BOOKSTALL

Publisher name	1997	1998	1999	2000	total
A	Rs.1000.00	Rs.1100.00	Rs.1300.00	Rs.800.00	
B	Rs.1500.00	Rs.700.00	Rs.1000.00	Rs.2000.00	
C	Rs.700.00	Rs.900.00	Rs.1500.00	Rs.600.00	
D	Rs.1200.00	Rs.500.00	Rs.200.00	Rs.1100.00	
E	Rs.800.00	Rs.1000.00	Rs.3000.00	Rs.560.00	

- (a) Compute the total revenue earned.  
 (b) Plot the line chart to compare the revenue of all publisher for 4 years.  
 (c) Chart Title should be \_Total Revenue of sam's Bookstall (1997-2000)'

(d) Give appropriate categories and value axis title.

10. Generate 25 random numbers between 0 & 100 and find their sum, average and count. How many no. are in range 50-60.

### **Suggested Readings**

1. Bittu Kumar (2015). Microsoft Office 2010. VS Publishers, New Delhi
2. Ramesh Bangia (2015). Learning Microsoft Office 2010, UBS Publishers.
3. Peter Weverka (2010) Office 2010 All-in-One For Dummies, Wiley Publishing Inc.



**18AEC101****ENVIRONMENTAL STUDIES****Semester – I  
4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100  
End Semester Exam : 3 Hours****Course Objectives (CO)**

- To create the awareness about environmental problems among people.
- To develop an attitude of concern for the environment.
- To motivate public to participate in environment protection and improvement.
- To understand the transnational character of environmental problems and ways of addressing them, including interactions across local to global scales.
- To apply systems concepts and methodologies to analyze and understand interactions between social and environmental processes.
- To reflect critically about their roles and identities as citizens, consumers and environmental actors in a complex, interconnected world.

**Course Outcomes (COs)**

1. Master core concepts and methods from ecological and physical sciences and their application in environmental problem solving.
2. Master core concepts and methods from economic, political, and social analysis as they pertain to the design and evaluation of environmental policies and institutions.
3. Appreciate the ethical, cross-cultural, and historical context of environmental issues and the links between human and natural systems.
4. Understand the transnational character of environmental problems and ways of addressing them, including interactions across local to global scales.
5. Apply systems concepts and methodologies to analyze and understand interactions between social and environmental processes.
6. Reflect critically about their roles and identities as citizens, consumers and environmental actors in a complex, interconnected world.

**Unit I - INTRODUCTION**

Environment Definition, Course Objectives and importance, components, Ecosystem Definition, Concept, Course Objectives, importance, Structure and functions of ecosystem. Energy flow, Ecological succession Food chains and food webs. Classification of ecosystem.

**Unit II - NATURAL RESOURCES - RENEWABLE AND NON-RENEWABLE RESOURCES**

Natural resources and associated problems. Forest resources, Water resources, Mineral resources, Food resources, Energy resources, Land resources: Use and over-utilization, exploitation. Role of an individual in conservation of natural resources. Equitable use of resources for sustainable lifestyles. Ill-effects of fire works.

**Unit III - BIODIVERSITY AND ITS CONSERVATION**

Introduction, definition: genetic, species and ecosystem diversity. Biogeographical classification of India. Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values. Biodiversity at global, National and local levels. India as a mega-diversity nation. Hot-spots of biodiversity. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts. Endangered and endemic species of India. Conservation of biodiversity: in-situ and ex-situ conservation of biodiversity.

#### **Unit IV - ENVIRONMENTAL POLLUTION**

Definition, Causes, effects and control measures of Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, Nuclear hazards, Solid waste management: Causes, effects and control measures of urban and industrial wastes. Role of an individual in prevention of pollution. Pollution case studies. Disaster management: Foods, earthquake, cyclone and landslides.

#### **Unit V - SOCIAL ISSUES AND THE ENVIRONMENT**

From unsustainable to sustainable development. Urban problems related to energy. Water conservation, rain water harvesting, watershed management. Resettlement and rehabilitation of people; its problems and concerns. Environmental ethics: Issues and possible solutions. Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Case studies. Wasteland reclamation. Consumerism and waste products. Environment Protection Act. Air (Prevention and Control of Pollution) Act. Water (Prevention and Control of Pollution) Act. Wildlife Protection Act. Forest Conservation Act. Issues involved in enforcement of environmental legislation. Public awareness. Population growth, variation among nations. Population explosion—Family Welfare Programme. Environment and human health. Human rights. Value education. HIV/AIDS. Women and Child Welfare. Role of Information Technology in environment and human health.

#### **Suggested Readings**

1. D.D.Mishra, (2010). Fundamental Concepts in Environmental Studies. S.Chand & Company Pvt. Ltd., New Delhi.
2. R. Rajagopalan, (2016) Environmental Studies: From Crisis to Cure, Oxford University Press
3. Tripathy. S.N., & Sunakar Panda. (2004). Fundamentals of Environmental Studies (2<sup>nd</sup> ed.) . Vrianda Publications Private Ltd, New Delhi.
4. Arvind Kumar. (2004). A Textbook of Environmental Science. APH Publishing Corporation, New Delhi.
5. Verma, P.S., & Agarwal V.K. (2001). Environmental Biology (Principles of Ecology) . S.Chand and Company Ltd, New Delhi.
6. Anubha Kaushik., & Kaushik, C.P. (2004). Perspectives in Environmental Studies. New Age International Pvt. Ltd. Publications, New Delhi.
7. Singh, M.P., Singh, B.S., & Soma, S. Dey. (2004). Conservation of Biodiversity and Natural Resources. Daya Publishing House. New Delhi.
8. Daniel, B. Botkin., & Edward, A. Keller. (1995). Environmental Science John Wiley and Sons, Inc., New York.

9. Uberoi, N.K. (2005). Environmental Studies. Excel Books Publications, New Delhi.

**பகுதி - I தமிழ்ப் பாடத்திட்டம் (2018 - 2019)**  
**இரண்டாம் பருவம்**  
**(இளநிலை அறிவியல் பட்ட வகுப்புகளுக்குரியது)**  
**(For I-UG Science Degree Classes) 18LSU201**

**பாடத்திட்டப் பொதுநோக்கம்**

- கற்றல் வழி சிந்தனைத் திறனையும், கருத்து வெளிப்பாட்டுத் திறனையும், மேம்படுத்துதல்.
- ஆய்வுநோக்கை மேம்படுத்துதல்.
- இலக்கியங்கள் உணர்த்தும் வாழ்வின் நுட்பமான பகுதிகளை உணர்த்துதல்.
- மனித மனத்தினைப் பக்குவப்படுத்துதலில் இலக்கியம் தரும் பங்கினை உணர்த்துதல்.
- வளர்ந்து வரும் சமூகத்தில் அறஉணர்வு, பண்பாடு போன்றவை குறித்து அறிவூட்டல்.
- அரசுத் தேர்வுகளுக்கு மாணவர்களை ஆயத்தமாக்குதல்.

**பாடத்திட்டப் பயன் விளைவு**

- இந்திய குடியுரிமைப் பணி முதலான போட்டித் தேர்வுகளில், விருப்பப் பாடமாக இந்திய குடியுரிமைப் பணி முதலான போட்டித் தேர்வுகளில், விருப்பப் பாடமாக இடம்பெறுகின்ற, 'தமிழ் இலக்கிய வரலாறு' குறித்த முழுமையான அறிமுகம் பெற்றிருத்தல்.
- கல்வெட்டியல், ஓலைச்சுவடியியல் மற்றும் தொல்லியல் சார்ந்த ஆவணத் தேடலுக்குரிய ஆய்வுமனப்பான்மையுடன், இலக்கியங்களை அணுகுதல்.
- தமிழின் வளர்ச்சித் துறையாகிய, 'அறிவியல் தமிழ்' ; 'இணைய தமிழ்' குறித்த பன்னோக்கு அணுகுமுறையிலான ஆய்வுச் சிந்தனை மேம்பாடு.
- வேலைவாய்ப்புக்குரிய சுயதிறன் மேம்பாட்டுடன், படைப்பாக்கத்திறன் மேம்பாடும் பெற்றிருத்தல் .
- சமுதாய மற்றும் வாழ்வியல் மதிப்புகளைப் பேணுவதற்குக் கருவியாக இலக்கியங்களை நாடுகின்ற மனப்பான்மை வளர்ச்சி.
- மொழிபெயப்புத் துறைசார்ந்த வேலைவாய்ப்புத் திறன் பெற்றிருத்தல்

பகுதி - I, தமிழ்

பருவம் II

18LSU201 :

தமிழ் இரண்டாம் தாள் 4-H,4-C  
(இளநிலை அறிவியல் பட்ட வகுப்புகளுக்குரியது)

அலகு - I : பக்தி இலக்கியம்

(10 மணிநேரம்)

சைவ, வைணவ இலக்கியங்கள் - தோற்றம் ,வளர்ச்சி, வரலாறு.

1. சைவம் - பெரியபுராணம் - திருமூலநாயனார் புராணம்.

2. வைணவம் - பெரியாழ்வார் திருமொழி: 10 பாடல்கள்.

அலகு - II : சங்க இலக்கியம்

:

(15 மணிநேரம்)

சங்க இலக்கியங்கள் அறிமுகம்

அ). எட்டுத்தொகை

நற்றிணை : பிரசம் கலந்த - பாலை -110

குறுந்தொகை : கருங்கட்டாக் கலை - குறிஞ்சி- 69

ஐங்குறுநூறு : நெய்தல்-தொண்டிப்பத்து:

திரைஇமிழ் இன்னிசை-171

பதிற்றுப்பத்து : சிதைந்தது மன்ற - 27

பரிபாடல்: பரிபாடல் திரட்டு-மதுரை நகர்ச்சிறப்பு -

உலகம் ஒரு நிறையாத்தான்-6, மாயோன் கொப்பூழ்-7, செய்யாட்கு

இழைத்த-9, கார்த்திகை காதில்-10, ஈவாரைக் கொண்டாடி-11.

கலித்தொகை : சுடர்தொடிக் கேளாய்: குறிஞ்சிக்கலி- 36

அகநானூறு : அன்னாய் வாழி வேண்டன்னை - குறிஞ்சி - 48

புறநானூறு : யாதும் ஊரே யாவருங் கேளிர் -பொதுவியல்- 192

ஆ). பத்துப்பாட்டு

திருமுருகாற்றுப்படை - பழமுதிர்ச்சோலையின் சிறப்பு

முருகன் இருப்பிடங்கள் - 'சிறுதினை மலரொடு' என்பதிலிருந்துதொடங்கி,

'அறிந்தவாறே' என்பது வரையிலான தொடர்கள்: 218-249.

முருகன் அருள்புரிதல் - 'தெய்வம் சான்ற' என்பதிலிருந்து தொடங்கி, 'நல்குமதி'

என்பது வரையிலான தொடர்கள்: 286-295.

அலகு - III : காப்பியம்

(6 மணிநேரம்)

சிலப்பதிகாரம்:

மங்கல வாழ்த்துப் பாடல்: (21-29) - கண்ணகியின் சிறப்பு:

நாகநீள் நகரொடு' என்பதிலிருந்து தொடங்கி,

'கண்ணகி என்பாண் மன்னோ' என்பது வரையிலான தொடர்கள்.

**நடுகற்காதை: (207-234) - சேரன் செங்குட்டுவன் கண்ணகிக்குக் கோயில் எடுத்தல்:**

'அருந்திறலரசர்' என்பதிலிருந்து தொடங்கி, 'மன்னவரேறென்' என்பது வரையிலான தொடர்கள்.

**வாழ்த்துக்காதை: (482-485) - செங்குட்டுவனுக்குக் கண்ணகி காட்சியளித்தல்:**

'என்னை' என்பதிலிருந்து தொடங்கி, 'விசும்பில் தோன்றுமால்' என்பது வரையிலான தொடர்கள்.

**வழக்குரை காதை: பத்தினிப் பெண்டிர் எழுவர் கதை: 'நீர்வார் கண்ணை'**  
என்பதிலிருந்து தொடங்கி, 'புகாரென் பதியே' என்பது வரையிலான தொடர்கள்.

**வஞ்சினமாலை: 'வன்னி மரமும்'** என்பதிலிருந்து தொடங்கி, 'பதிப்பிறந்தேன்' என்பது வரையிலான தொடர்கள்.

**அலகு – IV : சிறுகதை**

**(10 மணிநேரம்)**

1. குளத்தங்கரை அரசமரம் – வ.வே.சு.ஐயர்
2. காட்டில் ஒரு மான் - அம்பை
3. நாற்காலி – கி.ராஜநாராயணன்
4. நகரம் – சுஜாதா

**அலகு- V : மொழிப்பயிற்சி**

**(7 மணிநேரம்)**

படைப்பிலக்கியப் பயிற்சிகள் (கதை, கவிதை, கட்டுரை, உரைநடை)

மொழிபெயர்ப்பு [ ]

**பாட நூல்: கற்பகச்சோலை – தமிழ் ஏடு.** கற்பகம் பல்கலைக்கழகத் தமிழ்த் துறை வெளியீடு.

**18ENU201****ENGLISH****Semester – II  
4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100  
End Semester Exam : 3 Hours****Course Objectives:**

- To train students to acquire proficiency in English.
- To explore different genres of literature and learning grammar.
- To provide aesthetic pleasure through literature.
- To inculcate moral values through literature.
- To develop ethical values.
- To give basic grammar knowledge.

**Course Outcomes:**

- Develop the knowledge of interpersonal skills.
- Establish and maintain social relationships.
- Genres of literature will give moral values of life.
- Develop communication skills in business environment
- Communication skills will get developed.
- Develop to have language competence.

**Unit I - PROSE**

1. Morals in the Indian Context - Francis Nicholas Chelliah
2. How Comic Books help us to relive our Childhood - Benoit Peeters
3. Let's Do What India Needs From Us - Dr.A.P.J. Abdul Kalam

**Unit II - POEM**

1. The Stolen Boat - William Wordsworth
2. Telephone Conversation- Wole Soyinka
3. A River - A.K. Ramanujan

**Unit III - SHORT STORIES**

1. Rapunzel - Brothers Grimm
2. The Ant and The Grasshopper- W. Somerset Maugham
3. The Nightingale and the Rose - Oscar Wilde.

**Unit IV - DRAMA**

1. The Merchant of Venice- Act 4-Scene 1
2. The Death Trap- Saki

**Unit V - GRAMMAR AND COMPOSITION**

**GRAMMAR** : 1. Tenses  
2. Articles

3. Auxiliaries (Primary and Modal)
4. Tag Questions

**Composition:**

1. Reading to Comprehend
2. Letter Writing
3. Resume Writing
4. General Essay

**Prescribed Text:** Reminisce, Published by the Department of English, Karpagam Academy of Higher Education.

**Suggested Reading:** Hewings Martin, 1999 Advanced English Grammar, Cambridge University Press.



**18ITU201****PROGRAMMING IN JAVA****Semester – II  
4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100  
End Semester Exam : 3 Hours****Course Objectives (CO)**

- To understand the fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
- To understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
- To use the Java SDK environment to create, debug and run simple Java programs.
- To use Java in various technologies in different platforms.
- To understand the fundamental of Packages and access modifiers and interface in java.
- To understand the fundamental of Exception Handling and AWT component and AWT classes.

**Course Outcomes (COs)**

After completion of this course, the students will be able to

1. Obtain knowledge of the structure and model of the Java programming language.
2. Use the Java programming language for various programming technologies (understanding)
3. Develop software in the Java programming language (application)
4. Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements (analysis)
5. Use certain technologies by implementing them in the Java programming language to solve the given problem (synthesis)
6. Choose an engineering approach to solving problems, starting from the acquired knowledge of programming and knowledge of operating systems. (evaluation)

**Unit I - INTRODUCTION TO JAVA**

Java Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords Data Types, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Course Objectives, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods)

**Unit II - ARRAYS, STRINGS AND I/O**

Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character

streams, Reading/Writing from console and files. **Object-Oriented Programming Overview** Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

### **Unit III - INHERITANCE**

Inheritance: (Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes), Interfaces and Packages, Extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

### **Unit IV - EXCEPTION HANDLING AND DATABASE CONNECTIVITY**

Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

### **Unit V - JAVA APPLETS**

Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds. Event Handling Mechanisms, Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls, Swing components of Java Foundation Classes such as labels, buttons, textfields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts. Overview of servlets.

### **Suggested Readings**

1. Ken Arnold., James Gosling., & David Homes. (2005). The Java Programming Language (4th ed.).
2. James Gosling., Bill Joy., Guy, L. Steele Jr., Gilad Bracha., & Alex Buckley. (2014). The Java Language Specification, Java SE (8 ed.). Addison Wesley.
3. Joshua Bloch. (2008). Effective Java (2nd ed.). Addison-Wesley.
4. Cay, S. Horstmann., Gary Cornell. (2012). Core Java 2 Volume 1 (9th ed.). . Prentice Hall, New Delhi.
5. Cay, S. Horstmann., Gary Cornell. (2013). Core Java 2 Volume 2 - Advanced Features (9th ed.). Printice Hall, New Delhi.
6. Bruce Eckel. (2002). Thinking in Java (3rd ed.). PHI, New Delhi
7. Balaguruswamy, E. (2009). Programming with Java (4th ed.). McGraw Hill, New Delhi.
8. Paul Deitel., & Harvey Deitel. (2011). Java: How to Program (10th ed.). Prentice Hall, New Delhi.
9. David, J. Eck. (2009). Introduction to Programming Using Java. CreateSpace Independent Publishing Platform, New Delhi.
10. John , R. Hubbard. (2004). Programming with JAVA, Schaum's Series, (2nd ed.).

### **Web Sites**

1. [java.sun.com/docs/books/tutorial/](http://java.sun.com/docs/books/tutorial/)
2. [www.java.net/](http://www.java.net/)

18ITU202

DISCRETE STRUCTURES

Semester – II  
4H – 4C

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To learn the basic concepts of sets, types of sets, functions and relations
- To understand about Pigeonhole principle, Permutation and Combination, Mathematical Induction
- To solve the problems using Recurrence relations and generating functions.
- To know the basic concepts of Logical Connectives, Graphs and Trees.
- To express ideas using mathematical notation
- To solve problems with the help of tools of mathematical analysis.

**Course Outcomes (COs)**

On successful completion of the course, students will be able to

1. Familiar with elementary algebraic set theory.
2. Acquire a fundamental understanding of the core concepts in growth of functions.
3. Describe the method of recurrence relations.
4. Get wide knowledge about graphs and trees
5. Initiate to knowledge from inference theory
6. Solve problems with the help of tools of mathematical analysis

**Unit I**

**Sets:** Introduction, Sets, finite and infinite sets, uncountably infinite sets, functions, relations, properties of binary relations, closure, partial ordering relations, counting, Pigeonhole principle, Permutation and Combination, Mathematical Induction, Principle of inclusion and Exclusion.

**Unit II**

**Growth of Functions:** Asymptotic Notations, Summation formulas and properties, Bounding Summations, approximation by Integrals

**Unit III**

**Recurrences:** Recurrence relations, generating functions, linear recurrence relations with constant coefficients and their solution, Substitution Method, recurrence trees, Master theorem.

**Unit IV**

**Graph Theory:** Basic terminology, models and types, multigraphs and weighted graphs, graph representation, graph isomorphism, connectivity, Euler and Hamiltonian Paths and circuits, Planar graphs, graph coloring, trees, basic terminology and properties of trees, introduction to Spanning trees

**Unit V**

**Propositional Logic:** Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory.

**Suggested Readings**

1. Kenneth Rosen. (2012). Discrete Mathematics and Its Applications (7th ed.). New Delhi: McGraw Hill.
2. Tremblay , J .P. , &Manohar, R. (1997). Discrete Mathematical Structures with Applications to Computer Science. New Delhi: McGraw-Hill Book Company.
3. Coremen, T.H., Leiserson, C.E. , & R. L. Rivest. (2009). Introduction to algorithms, (3rd ed.). New Delhi: Prentice Hall on India.
4. Albertson, M. O.,& Hutchinson, J. P. (1988). Discrete Mathematics with Algorithms . New Delhi: John wiley Publication.
5. Hein, J. L. (2009). Discrete Structures, Logic, and Computability( 3rd ed.). New Delhi: Jones and Bartlett Publishers.
6. Hunter, D.J. (2017). Essentials of Discrete Mathematics. New Delhi: Jones and Bartlett Publishers.



**18ITU203****COMPUTER NETWORKS AND INTERNET  
TECHNOLOGIES****Semester – II  
4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100  
End Semester Exam : 3 Hours****Course Objectives (CO)**

- To study the basics of Computer Networks.
- To study and compare various Network architectures and fundamental protocols.
- To learn about networking protocol and OSI model.
- To learn various transmission media.
- To understand the topologies of networks, layered architecture (OSI and TCP/IP) and protocol suites, to learn the language of HTML, DHTML, XML and PHP.
- To understand the principles of creating an effective web page, to develop skills in analyzing the usability of a website.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Independently understand basic computer network technology.
2. Understand and explain Data Communications System and its components.
3. Identify the different types of network topologies and protocols, enumerate the layers of the OSI model and TCP/IP.
4. Employ fundamental computer theory to basic programming techniques, gain the skills and project-based experience needed for entry into web design and development careers.
5. Develop awareness and appreciation of the many ways that people access the web, and will be able to create standards-based websites that can be accessed by the full spectrum of web access technologies
6. Select and apply markup languages for processing, identifying, and presenting of information in web pages, create and manipulate web media objects using editing software.

**Unit I - COMPUTER NETWORKS**

Introduction to computer network, data communication, components of data communication, data transmission mode, data communication measurement, LAN, MAN, WAN, wireless LAN, internet, intranet, extranet. **Network Models:** Client/ server network and Peer-to-peer network, OSI, TCP/IP, 8L layers and functionalities.

**Unit II - TRANSMISSION MEDIA AND LAN TOPOLOGIES**

Introduction, Guided Media: Twisted pair, Coaxial cable, 4L Optical fiber. Unguided media: Microwave, Radio frequency propagation, Satellite. **LAN Topologies:** Ring, bus, star, mesh and tree topologies. Network Devices: NIC, repeaters, hub, bridge, switch, gateway and router.

**Unit III - INTERNET TERMS AND APPLICATIONS**

Internet Terms: Web page, Home page, website, internet browsers, URL, Hypertext, 2L ISP, Web server, download and upload, online and offline. **Internet Applications:** www, telnet, ftp, e-mail, social networks, search engines, 6L Video Conferencing, e-Commerce, m-Commerce, VOIP, blogs.

**Unit IV - INTRODUCTION TO WEB DESIGN**

Introduction to hypertext markup language (html) 16L Document type definition, creating web pages, lists, hyperlinks, tables, web forms, inserting images, frames, hosting options and domain name registration. **Customized Features:** Cascading style sheet (css) for text formatting and other manipulations.

**Unit V - JAVASCRIPT FUNDAMENTALS**

Data types and variables, functions, methods and events, 14L controlling program flow, JavaScript object model, built-in objects and operators.

**Suggested Readings**

1. Larry L.Peterson & Bruce S.Davie (2011). Computer Networks A System Approach, Morgan Kaufmann Publishers.
2. Andrew S. Tanenbaum, David J. Wetherall Computer Networks (5th Edition), PHI, 2010
3. B. A. Forouzan, Data Communication and Networking , TMH,2003.
4. D.R. Brooks, An Introduction to HTML and Javascript for Scientists and Engineers, Springer W. Willard,2009
5. HTML A Beginner's Guide, Tata McGraw-Hill Education, 2009.
6. J. A. Ramalho, Learn Advanced HTML 4.0 wit

**Web Sites**

1. <https://developer.mozilla.org/en-US/docs/Web>
2. <https://www.w3schools.com>
3. [http://en.wikipedia.org/wiki/script\\_language](http://en.wikipedia.org/wiki/script_language)
4. <https://css-tricks.com>

---

<b>18ITU211</b>	<b>PROGRAMMING IN JAVA - PRACTICAL</b>	<b>Semester – II 4H – 2C</b>
-----------------	--	----------------------------------

---

**Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To understand the fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
- To understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
- To use the Java SDK environment to create, debug and run simple Java programs.
- To use Java in various technologies in different platforms.
- To understand the fundamental of Packages and access modifiers and interface in java.
- To understand the fundamental of Exception Handling and AWT component and AWT classes.

**Course Outcomes (COs)**

After completion of this course, the students will be able to

1. Obtain knowledge of the structure and model of the Java programming language.
2. Use the Java programming language for various programming technologies (understanding)
3. Develop software in the Java programming language (application)
4. Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements (analysis)
5. Use certain technologies by implementing them in the Java programming language to solve the given problem (synthesis)
6. Choose an engineering approach to solving problems, starting from the acquired knowledge of programming and knowledge of operating systems. (evaluation)

**List of Programs**

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of length in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and StringBuffer class like setCharAt(), setLength(), append(), insert(), concat() and equals().
9. Write a program to create a —distance class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
10. Modify the —distance class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference



variable to another object reference variable. Further create a third object which is a clone of the first object.

11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions (from lower to higher data type)
12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
13. Write a program to show the use of static functions and to pass variable length arguments in a function.
14. Write a program to demonstrate the concept of boxing and unboxing.
15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate fibonacci series is given in a different file belonging to the same package.
17. Write a program that creates/illustrates different levels of protection in classes/subclasses belonging to same package or different packages
18. Write a program —DivideByZero that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
20. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
21. Write a program to demonstrate priorities among multiple threads.
22. Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).
23. Write a program to create URL object, create a URLConnection using the openConnection() method and then use it to examine the different components of the URL and content.
24. Write a program to implement a simple datagram client and server in which a message that is typed into the server window is sent to the client side where it is displayed.
25. Write a program that creates a Banner and then creates a thread to scroll the message in the banner from left to right across the applet's window.
26. Write a program to get the URL/location of code (i.e. java code) and document (i.e. html file).
27. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed(), mouseReleased() and mouseDragged().
28. Write a program to demonstrate different keyboard handling events.
29. Write a program to generate a window without an applet window using main() function.
30. Write a program to demonstrate the use of push buttons.

### Suggested Readings

1. James Gosling., Bill Joy., Guy, L. Steele Jr., Gilad Bracha., & Alex Buckley. (2014). The Java

- Language Specification, Java SE (8 ed.). Addison Wesley.
2. Joshua Bloch. (2008). Effective Java (2nd ed.). Addison-Wesley.
  3. Cay, S. Horstmann., Gary Cornell. (2012). Core Java 2 Volume 1 (9th ed.). . Prentice Hall, New Delhi.
  4. Cay, S. Horstmann., Gary Cornell. (2013). Core Java 2 Volume 2 - Advanced Features(9th ed.). Printice Hall, New Delhi.
  5. Balaguruswamy, E. (2009). Programming with Java (4th ed.). McGraw Hill, New Delhi.
  6. Paul Deitel., & Harvey Deitel. (2011). Java: How to Program (10th ed.). Prentice Hall, New Delhi.

**Web Sites**

1. [java.sun.com/docs/books/tutorial/](http://java.sun.com/docs/books/tutorial/)
2. [www.en.wikipedia.org/wiki/Java](http://www.en.wikipedia.org/wiki/Java)
3. [www.java.net/](http://www.java.net/)

**18ITU212****DISCRETE STRUCTURES - PRACTICAL****Semester – II  
3H – 2C**

**Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Internal : 40 External : 60 Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To learn the basic concepts of sets, types of sets, functions and relations
- To understand about Pigeonhole principle, Permutation and Combination, Mathematical Induction
- To solve the problems using Recurrence relations and generating functions.
- To know the basic concepts of Logical Connectives, Graphs and Trees.
- To express ideas using mathematical notation
- To solve problems with the help of tools of mathematical analysis.

**Course Outcomes (COs)**

On successful completion of the course, students will be able to

1. Familiar with elementary algebraic set theory.
2. Acquire a fundamental understanding of the core concepts in growth of functions.
3. Describe the method of recurrence relations.
4. Get wide knowledge about graphs and trees
5. Initiate to knowledge from inference theory
6. Solve problems with the help of tools of mathematical analysis

**List of Programs**

1. Write a C Program to find the number of subsets of a set contains n elements.
2. Write a C Program to find transitive closure of a relation.
3. Write a C Program to prove  
 $1/(1*2) + 1/(2*3) + \dots + 1/(n(n+1)) = n/(n+1)$
4. Write a C Program to to perform the sum =  $1 + (1+2) + (1+2+3) + \dots + (1+2+\dots+n)$
5. Write a C program to print Fibonacci series till Nth term using recursion
6. Write a C program in c to calculate factorial of a number using recursion
7. Write a C Program to find a minimum spanning tree using Prim's algorithm
8. Write a C program to find the shortest path with the lower cost in a graph using Dijkstra's Algorithm
9. Write a C Program to construct the truth table for the following formula.  
 (i)  $P \wedge Q \wedge \neg R$     (ii)  $P \wedge \neg Q \wedge R$     (iii)  $P \wedge Q \wedge \neg R$

10. Write a C Program to prove De – Morgan’s law.

### **Suggested Readings**

1. Kenneth Rosen. (2006). Discrete Mathematics and Its Applications (6th ed.). McGraw Hill, New Delhi.
2. Tremblay , J .P. , & Manohar, R. (1997). Discrete Mathematical Structures with Applications to Computer Science. McGraw-Hill Book Company, New Delhi.
3. Coremen, T.H., Leiserson, C.E. , & R. L. Rivest. (2009). Introduction to algorithms, (3rd ed.). Prentice Hall on India, New Delhi.
4. Albertson, M. O.,& Hutchinson, J. P. (1988). Discrete Mathematics with Algorithms .: John wiley Publication, New Delhi.
5. Hein, J. L. (2009). Discrete Structures, Logic, and Computability( 3rd ed.). Jones and Bartlett Publishers, New Delhi.
6. Hunter, D.J. (2008). Essentials of Discrete Mathematics. Jones and Bartlett Publishers, New Delhi.

**18ITU213****COMPUTER NETWORKS AND INTERNET  
TECHNOLOGIES - PRACTICAL****Semester – II  
3H – 2C****Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Internal : 40 External : 60 Total: 100  
End Semester Exam : 3 Hours****Course Objectives (CO)**

- To study the basics of Computer Networks.
- To study and compare various Network architectures and fundamental protocols.
- To learn about networking protocol and OSI model.
- To learn various transmission media.
- To understand the topologies of networks, layered architecture (OSI and TCP/IP) and protocol suites, to learn the language of HTML, DHTML, XML and PHP.
- To understand the principles of creating an effective web page, to develop skills in analyzing the usability of a website.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Independently understand basic computer network technology.
2. Understand and explain Data Communications System and its components.
3. Identify the different types of network topologies and protocols, enumerate the layers of the OSI model and TCP/IP.
4. Employ fundamental computer theory to basic programming techniques, gain the skills and project-based experience needed for entry into web design and development careers.
5. Develop awareness and appreciation of the many ways that people access the web, and will be able to create standards-based websites that can be accessed by the full spectrum of web access technologies
6. Select and apply markup languages for processing, identifying, and presenting of information in web pages, create and manipulate web media objects using editing software.

**List of Programs**

1. Create HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.
2. Create HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking
3. Create HTML document with Table


4. Create Form with Input Type, Select and Text Area in HTML.
5. Create an HTML containing Roll No., student's name and Grades in a tabular form.
6. Create an HTML document (having two frames) which will appear as follows

<p>About</p> <p>Department 1</p> <p>Department 2</p> <p>Department 3</p>	<p>This frame would show the contents according to the link clicked by the user on the left frame.</p>
--	--

7. Create an HTML document containing horizontal frames as follows

Department Names (could be along with Logos)
Contents according to the Link clicked

8. Create a website of 6 – 7 pages with different effects as mentioned in above problems.
9. Create HTML documents (having multiple frames) in the following three formats

Frame1
Frame2

Frame1	
Frame2	Frame3

10. Create a form using HTML which has the following types of controls:
  - V. Text Box
  - VI. Option/radio buttons
  - VII. Check boxes
  - VIII. Reset and Submit buttons

List of Practicals using Javascript : Create event driven program for following:

11. Print a table of numbers from 5 to 15 and their squares and cubes using alert.
12. Print the largest of three numbers. 81
13. Find the factorial of a number n.
14. Enter a list of positive numbers terminated by Zero. Find the sum and average of

these numbers.

15. A person deposits Rs 1000 in a fixed account yielding 5% interest. Compute the amount in the account at the end of each year for n years.
16. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.

### **Suggested Readings**

1. Andrew S. Tanenbaum, David J. Wetherall Computer Networks (5th Edition), PHI, 2010
2. B. A. Forouzan, Data Communication and Networking , TMH, 2003.
3. D.R. Brooks, An Introduction to HTML and Javascript for Scientists and Engineers, Springer W. Willard, 2009
4. HTML A Beginner's Guide, Tata McGraw-Hill Education, 2009.
5. J. A. Ramalho, Learn Advanced HTML 4.0 wit

**18ITU301****DATA STRUCTURES****Semester – III  
4H – 4C**

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100  
End Semester Exam : 3 Hours**

---

**Course Objectives (CO)**

- To teach efficient storage mechanisms of data for an easy access.
- To design and implementation of various basic and advanced data structures.
- To introduce various techniques for representation of the data in the real world.
- To develop application using data structures.
- To teach the concept of protection and management of data.
- To improve the logical ability

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Choose appropriate data structure as applied to specified problem definition.
2. Handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
3. Identify different parameters to analyze the performance of an algorithm.
4. Apply concepts learned in various domains like DBMS, compiler construction etc.
5. Use linear and non-linear data structures like stacks, queues, linked list etc.
6. Illustrate various technique to for searching, Sorting and hashing

**Unit I**

Arrays-Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation).Stacks Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack

**Unit II**

Linked Lists Singly, Doubly and Circular Lists (Array and Linked representation); Normal and Circular, representation of Stack in Lists; Self Organizing Lists; Skip Lists Queues, Array and Linked representation of Queue, De-queue, Priority Queues

**Unit III**

Trees - Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion , Recursive and Iterative Traversals on Binary Search Trees); Threaded Binary Trees (Insertion, Deletion, Traversals); Height-Balanced Trees (Various operations on AVL Trees).

**Unit IV**

Searching and Sorting,Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Insertion Sort, Shell Sort, Comparison of Sorting Techniques

**Unit V**



Hashing - Introduction to Hashing, Deleting from Hash Table, Efficiency of Rehash Methods, Hash Table Reordering, Resolving collision by Open Addressing, Coalesced Hashing, Separate Chaining, Dynamic and Extendible Hashing, Choosing a Hash Function, Perfect Hashing, Function.

### Suggested Readings

1. Adam Drozdek. (2012). Data Structures and algorithm in C++(3<sup>rd</sup> ed.). Cengage Learning.
2. Sartaj Sahni. (2011). Data Structures, Algorithms and applications in C++(2<sup>nd</sup> ed.). Universities Press.
3. Aaron, M. Tenenbaum., Moshe, J. Augenstein., & Yedidiah Langsam. (2009). Data Structures Using C and C++(2<sup>nd</sup> ed.). PHI.
4. Robert, L. Kruse. (1999). Data Structures and Program Design in C++. Pearson.
5. D.S.Malik (2010). Data Structure using C++(2<sup>nd</sup> ed.). Cengage Learning,.
6. Mark Allen Weiss. (2011). Data Structures and Algorithms Analysis in Java (3<sup>rd</sup> ed.). Pearson Education.
7. Aaron M. Tenenbaum., Moshe, J. Augenstein., & Yedidiah Langsam. (2003). Data Structures Using Java. PHI.
8. Robert Lafore. (2003). Data Structures and Algorithms in Java(2<sup>nd</sup> ed.). Pearson/ Macmillan Computer Pub.
9. John Hubbard. (2009). Data Structures with JAVA(2<sup>nd</sup> ed.). McGraw Hill Education (India) Private Limited.
10. Goodrich, M., & Tamassia, R. (2013). Data Structures and Algorithms Analysis in Java(4<sup>th</sup> ed.). Wiley.
11. Herbert Schildt. (2014). Java The Complete Reference (English)(9<sup>th</sup> ed.). Tata McGraw Hill.
12. D. S.Malik, P.S.Nair (2003).Data Structures Using Java. .Course Technology.

### Web Sites

1. [http://en.wikipedia.org/wiki/Data\\_structure](http://en.wikipedia.org/wiki/Data_structure)
2. <http://www.cs.sunysb.edu/~skiena/214/lectures/>
3. [www.amazon.com/Teach-Yourself-Structures-Algorithms](http://www.amazon.com/Teach-Yourself-Structures-Algorithms)

**Semester – III****18ITU302****OPERATING SYSTEMS****4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100****End Semester Exam : 3 Hours****Course Objectives (CO)**

- To understand the main components of an OS & their functions.
- To study the process management and scheduling.
- To understand various issues in Inter Process Communication (IPC) and the role of OS in IPC. To understand the concepts and implementation Memory management policies and virtual memory.
- To understand the working of an OS as a resource manager, file system manager, process manager, memory manager and I/O manager and methods used to implement the different parts of OS
- To study the need for special purpose operating system with the advent of new emerging technologies
- To understand the structure and organization of the file system

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Describe the important computer system resources and the
2. Perform the role of operating system in their management policies and algorithms.
3. Understand the process management policies and scheduling of processes by CPU
4. Evaluate the requirement for process synchronization and coordination handled by operating system
5. Describe and analyze the memory management and its allocation policies.
6. Identify use and evaluate the storage management policies with respect to different storage management technologies, identify the need to create the special purpose operating system.

**Unit I**

**Introduction to Operating System:** Basic OS Functions-Resource Abstraction-Types of Operating Systems–Multiprogramming Systems-Batch Systems-Time Sharing Systems-Operating Systems for Personal Computers & Workstations-Process Control & Real Time Systems.

**Unit II**

**Operating System Organization:** Processor and user modes-Kernels-System Calls and System Programs. **Process Management:** System view of the process and resources- Process abstraction-Process hierarchy-Threads-Threading issues-Thread libraries-Process Scheduling-Non pre-emptive and Preemptive scheduling algorithms-Concurrent and processes-Critical Section-Semaphores-Methods for inter-process communication- Deadlocks.

**Unit III**

**Memory Management:** Physical and Virtual address space-Memory Allocation strategies – Fixed and Variable partitions-Paging-Segmentation-Virtual memory.

**Unit IV**

**File and I/O Management:** Directory structure-File operations-File Allocation methods- Device management.

**Unit V**

**Protection and Security:** Policy mechanism-Authentication-Internal aITUess Authorization.

**Suggested Readings**

1. A .Silberschatz, , P.B Galvin, G.Gagne (2008). Operating Systems Concepts, 8<sup>th</sup> ed.). John Wiley Publications.
2. A.S. Tanenbaum, (2007).Modern Operating Systems (3<sup>rd</sup> ed.). New Delhi: Pearson Education.
3. W. Stallings, ( 2008). Operating Systems, Internals & Design Principles (5<sup>th</sup> ed.). Prentice Hall of India.

**Web Sites**

1. [www.cs.columbia.edu/~nieh/teaching/e6118\\_s00/](http://www.cs.columbia.edu/~nieh/teaching/e6118_s00/)
2. [www.clarkson.edu/~jnm/cs644](http://www.clarkson.edu/~jnm/cs644)
3. [pages.cs.wisc.edu/~remzi/Classes/736/Fall2002/](http://pages.cs.wisc.edu/~remzi/Classes/736/Fall2002/)

---

<b>18ITU303</b>	<b>RELATIONAL DATABASE MANAGEMENT SYSTEMS</b>	<b>Semester – III</b> <b>4H – 4C</b>
-----------------	---	---

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal: 40    External: 60    Total: 100**  
**End Semester Exam: 3 Hours**

**Course Outcomes (COs)**

- To describe a sound introduction to the discipline of database management systems.
- To give a good formal foundation on the relational model of data and usage of Relational Algebra.
- To introduce the concepts of basic SQL as a universal Database language.
- To enhance knowledge to advanced SQL topics like embedded SQL, procedures connectivity through JDBC.
- To demonstrate the principles behind systematic database design approaches by covering conceptual design, logical design through normalization.
- To provide an overview of physical design of a database system, by discussing Database indexing techniques and storage techniques.

**Course Outcomes (COs)**

Upon completion of the course, students will be able to

1. Explain the features of database management systems and Relational database.
2. Design conceptual models of a database using ER modeling for real life applications and also construct queries in Relational Algebra.
3. Create and populate a RDBMS for a real life application, with constraints and keys, using SQL.
4. Retrieve any type of information from a data base by formulating complex queries in SQL.
5. Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.
6. Build indexing mechanisms for efficient retrieval of information from a database

**Unit I**

DBMS Definition, Characteristics of DBMS ,Application and advantages of DBMS, Instances , Schemas and Database States, Three Levels of Architecture , Data Independence, DBMS languages, Data Dictionary, Database Users, Data Administrators.

**Unit II**

Data Models, types and their comparison, Entity Relationship Model, Entity Types, Entity Sets, Attributes and its types, Keys, E-R Diagram, Data Integrity RDBMS –Concept, Components and Codd's rules.

**Unit III**

Relational Algebra (selection, projection, union, intersection, Cartesian product, Different types of join like theta join, equi-join, natural join, outer join)

Functional Dependencies, Good & Bad Decomposition, Anomalies as a database: A consequences of bad design, Normalization: 1NF, 2NF, 3NF, BCNF, 4NF 5NF.

#### **Unit IV**

Introduction to SQL: DDL, DML, and DCL statements, Creating Tables, Adding Constraints, Altering Tables, Update, Insert, Delete & various Form of SELECT- Simple, Using Special Operators for Data Access. Aggregate functions, Joining Multiple Tables (Equi Joins),Joining a Table to itself (self Joins) Functions.

Introduction to PL/SQL: Declaration section – executable command section : conditional logic, loops, CASE statements –

#### **Unit V**

Exception handling section: predefined and user defined exceptions. Triggers: definition – types: row level, statement level, before and after, instead of – syntax – enabling and disabling triggers - replacing and dropping triggers. Cursors – definition – open – fetch – close – cursor attributes- select for update – types : implicit, explicit. Procedures, Functions: Local and global – procedures vs functions – stored procedures, functions – create procedure syntax - create function syntax – calling procedures, functions. Replacing and dropping procedures, functions. Package header – package body – calling package members - Replacing and dropping package.

#### **Suggested Readings**

1. Bipin C. Desai.(2013). An Introduction to Database Systems, New Delhi: Galgotia Publications.
2. Rajiv chopra (2013). Database Management systems (3<sup>rd</sup> ed.). S.Chand publications.
3. Steven Feurstein, Bill Pribyl (2014). Oracle PL/SQL Programming (6<sup>th</sup> ed.). O ‘ Reilly Media.
4. Shio Kumar Singh (2011). Database Management Systems – Concepts, design and Applications (2<sup>nd</sup> ed.). New Delhi: Pearson Education.
5. Ivan Byross (2010). SQL, PL/SQL the Programming Language of Oracle Paperback. BPB Publications.
6. Rajeeb C. Chatterjee (2012). Learning Oracle SQL and PL/SQL: A simplified Guide. Prentice Hall of India.

#### **Web Sites**

1. <http://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm>
2. [www.databasedir.com](http://www.databasedir.com)
3. <http://plsql-tutorial.com/>

**Semester – III****18ITU304A****ANDROID PROGRAMMING****3H – 3C**

**Instruction Hours / week: L: 3 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To introduce Android platform and its architecture.
- To learn activity creation and Android UI designing.
- To be familiarized with Intent, Broadcast receivers and Internet services.
- To work with SQLite Database and content providers.
- To integrate multimedia, camera and Location based services in Android Application.
- To explore Mobile security issues.

**Course Outcomes (COs)**

Upon completion of this course, the students will able to

1. Describe Android platform, Architecture and features.
2. Design User Interface and develop activity for Android App.
3. Use Internet, Broadcast receivers and Internet services in Android App.
4. Design and implement Database Application and Content providers.
5. Use multimedia, camera and Location based services in Android App.
6. Discuss various security issues in Android platform

**Unit I**

**Introduction:** History of Android, Introduction to Android Operating Systems, Android Development Tools, Android Architecture.

**Unit II**

**Overview of object oriented programming using Java:** OOPs Concepts: Inheritance, Polymorphism, Interfaces, Abstract class, Threads, Overloading and Overriding, Java Virtual Machine.

**Unit III**

**Development Tools:** Installing and using Eclipse with ADT plug-in, Installing Virtual machine for Android sandwich/Jelly bean (Emulator), configuring the installed tools, creating a android project– Hello Word, run on emulator, Deploy it on USB-connected Android device.

**Unit IV**

**User Interface Architecture:** Application context, intents, Activity life cycle, multiple screen sizes.

**User Interface Design:** Form widgets, Text Fields, Layouts, Button control, toggle buttons, Spinners(Combo boxes),Images, Menu, Dialog.

**Unit V**

**Database:** Understanding of SQLite database, connecting with the database.

**Suggested Readings**

1. James, C. Sheusi.( 2013). Android application development for java rogrammers. Cengage Learning.

**Web Sites**

1. <http://www.developer.android.com>
2. <http://developer.android.com/about/versions/index.html>
3. <http://developer.android.com/training/basics/firstapp/index.html>
4. <http://docs.oracle.com/javase/tutorial/index.htm>(Available in the form of free downloadable ebooks also).
5. <http://developer.android.com/guide/components/activities.html>
6. <http://developer.android.com/guide/components/fundamentals.html>
7. <http://developer.android.com/guide/components/intents-filters.html>.
8. <http://developer.android.com/training/multiscreen/screensizes.html>
9. <http://developer.android.com/guide/topics/ui/controls.html>
10. <http://developer.android.com/guide/topics/ui/declaring-layout.html>
11. <http://developer.android.com/training/basics/data-storage/databases.html>

---

<b>18ITU304B</b>	<b>PROGRAMMING IN PYTHON</b>	<b>Semester – III</b>
		<b>3H – 3C</b>

---

**Instruction Hours / week: L: 3 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To Learn Syntax and Semantics and create Functions in Python.
- To Handle Strings and Files in Python.
- To Understand Lists, Dictionaries in Python.
- To Implement Object Oriented Programming concepts in Python
- To Build GUI applications
- To Write Python functions to facilitate code reuse.

**Course Outcomes (COs)**

Upon completion of the course, students will be able to

1. Develop algorithmic solutions to simple computational problems
2. Read, write, execute by hand simple Python programs.
3. Structure simple Python programs for solving problems.
4. Decompose a Python program into functions.
5. Represent compound data using Python lists, tuples, dictionaries.
6. Read and write data from/to files in Python Programs.

**Unit I**

**Algorithmic Problem Solving:** Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudocode, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi.

**Unit II**

**Data, Expressions, Statements :** Python interpreter and interactive mode; values and types: int, float, boolean, string, and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; modules and functions, function definition and use, flow of execution, parameters and arguments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.

**Unit III**

**Control Flow, Functions:** Conditionals: Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else); Iteration: state, while, for, break, continue, pass; Fruitful functions: return values, parameters, local and global scope, function composition, recursion; Strings: string slices, immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.



**Unit IV**

**Lists, Tuples, Dictionaries :** Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: selection sort, insertion sort, merge sort, histogram.

**Unit V**

**Files, Modules, Packages:** Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file.

**Suggested Readings**

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist``, 2nd edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016 (<http://greenteapress.com/wp/thinkpython/>)
2. Guido van Rossum and Fred L. Drake Jr, —An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.
3. John V Guttag, —Introduction to Computation and Programming Using Python``, Revised and expanded Edition, MIT Press , 2013
4. Robert Sedgewick, Kevin Wayne, Robert Dondero, —Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016.
5. Timothy A. Budd, —Exploring Python, Mc-Graw Hill Education (India) Private Ltd., 2015.
6. Kenneth A. Lambert, —Fundamentals of Python: First Programs, CENGAGE Learning, 2012.
7. Charles Dierbach, —Introduction to Computer Science using Python: A Computational ProblemSolving Focus, Wiley India Edition, 2013.
8. Paul Gries, Jennifer Campbell and Jason Montoyo, —Practical Programming: An Introduction to Computer Science using Python 3, Second edition, Pragmatic Programmers, LLC, 2013.

**Semester – III****18ITU311****DATA STRUCTURES - PRACTICAL****4H – 2C****Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100****End Semester Exam : 3 Hours****Course Objectives (CO)**

- To teach efficient storage mechanisms of data for an easy access.
- To design and implementation of various basic and advanced data structures.
- To introduce various techniques for representation of the data in the real world.
- To develop application using data structures.
- To teach the concept of protection and management of data.
- To improve the logical ability

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Choose appropriate data structure as applied to specified problem definition.
2. Handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
3. Identify different parameters to analyze the performance of an algorithm.
4. Apply concepts learned in various domains like DBMS, compiler construction etc.
5. Use linear and non-linear data structures like stacks, queues, linked list etc.
6. Illustrate various technique to for searching, Sorting and hashing

**List of Programs**

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.
2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.

8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.
11. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
12. (ii) WAP to display fibonacci series (i)using recursion, (ii) using iteration
13. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion
14. WAP to create a Binary Search Tree and include following operations in tree: (a) Insertion (Recursive and Iterative Implementation)  
(b) Deletion by copying  
(c) Deletion by Merging  
(d) Search a no. in BST  
(e) Display its preorder, postorder and inorder traversals Recursively  
(f) Display its preorder, postorder and inorder traversals Iteratively  
(g) Display its level-by-level traversals  
(h) Count the non-leaf nodes and leaf nodes  
(i) Display height of tree  
(j) Create a mirror image of tree  
(k) Check whether two BSTs are equal or not
15. WAP to convert the Sparse Matrix into non-zero form and vice-versa.
16. WAP to reverse the order of the elements in the stack using additional stack.
17. WAP to reverse the order of the elements in the stack using additional Queue.
18. WAP to implement Diagonal Matrix using one-dimensional array.
19. WAP to implement Lower Triangular Matrix using one-dimensional array.
20. WAP to implement Upper Triangular Matrix using one-dimensional array.
21. WAP to implement Symmetric Matrix using one-dimensional array.
22. WAP to create a Threaded Binary Tree as per inorder traversal, and implement operations like finding the successor / predecessor of an element, insert an element, inorder traversal.
23. WAP to implement various operations on AVL Tree.

### **Suggested Readings**

1. Adam Drozdek. (2012). Data Structures and algorithm in C++(3<sup>rd</sup> ed.). Cengage Learning.
2. Sartaj Sahni. (2011). Data Structures, Algorithms and applications in C++(2<sup>nd</sup> ed.). Universities Press.
3. Aaron, M. Tenenbaum., Moshe, J. Augenstein., & Yedidiah Langsam. (2009). Data Structures Using C and C++(2<sup>nd</sup> ed.). PHI.
4. Robert, L. Kruse. (1999). Data Structures and Program Design in C++. Pearson.
5. D.S.Malik (2010). Data Structure using C++(2<sup>nd</sup> ed.). Cengage Learning,.
6. Mark Allen Weiss. (2011). Data Structures and Algorithms Analysis in Java (3<sup>rd</sup> ed.). Pearson Education.
7. Aaron M. Tenenbaum., Moshe, J. Augenstein., & Yedidiah Langsam. (2003). Data Structures Using Java. PHI.
8. Robert Lafore. (2003). Data Structures and Algorithms in Java(2<sup>nd</sup> ed.). Pearson/ Macmillan Computer Pub.
9. John Hubbard. (2009). Data Structures with JAVA(2<sup>nd</sup> ed.). McGraw Hill Education (India) Private Limited.
10. Goodrich, M., & Tamassia, R. (2013). Data Structures and Algorithms Analysis in Java(4<sup>th</sup> ed.). Wiley.
11. Herbert Schildt. (2014). Java The Complete Reference (English)(9<sup>th</sup> ed.). Tata McGraw Hill.
12. D. S.Malik, P.S.Nair (2003).Data Structures Using Java. .Course Technology.

**Web Sites**

1. [http://en.wikipedia.org/wiki/Data\\_structure](http://en.wikipedia.org/wiki/Data_structure)
2. <http://www.cs.sunysb.edu/~skiena/214/lectures/>
3. [www.amazon.com/Teach-Yourself-Structures-Algorithms](http://www.amazon.com/Teach-Yourself-Structures-Algorithms)

**18ITU312****OPERATING SYSTEMS - PRACTICAL****Semester – III  
4H – 2C****Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100****End Semester Exam : 3 Hours****Course Objectives (CO)**

- To understand the main components of an OS & their functions.
- To study the process management and scheduling.
- To understand various issues in Inter Process Communication (IPC) and the role of OS in IPC. To understand the concepts and implementation Memory management policies and virtual memory.
- To understand the working of an OS as a resource manager, file system manager, process manager, memory manager and I/O manager and methods used to implement the different parts of OS
- To study the need for special purpose operating system with the advent of new emerging technologies
- To provide necessary skills for developing and debugging programs in UNIX environment.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Describe the important computer system resources and the
2. Perform the role of operating system in their management policies and algorithms.
3. Understand the process management policies and scheduling of processes by CPU
4. Evaluate the requirement for process synchronization and coordination handled by operating system
5. Describe and analyze the memory management and its allocation policies.
6. Identify, use and evaluate the storage management policies with respect to different storage management technologies, identify the need to create the special purpose operating system.

**List of Programs**

1. Write a program (using fork() and/or exec() commands) where parent and child execute:
  - a) same program, same code.
  - b) same program, different code.
  - c) before terminating, the parent waits for the child to finish its task.
2. Write a program to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information)
3. Write a program to report behaviour of Linux kernel including information on configured memory, amount of free and used memory. (memory information)

4. Write a program to print file details including owner access permissions, file access time, where file name is given as argument.
5. Write a program to copy files using system calls.
6. Write program to implement FCFS scheduling algorithm.
7. Write program to implement Round Robin scheduling algorithm.
8. Write program to implement SJF scheduling algorithm.
9. Write program to implement non-preemptive priority based scheduling algorithm.
10. Write program to implement preemptive priority based scheduling algorithm.
11. Write program to implement SRJF scheduling algorithm.
12. Write program to calculate sum of n numbers using thread library.
13. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

**Suggested Readings**

1. A .Silberschatz, , P.B Galvin, G.Gagne (2008). Operating Systems Concepts, 8<sup>th</sup> ed.). John Wiley Publications.
2. A.S. Tanenbaum, (2007).Modern Operating Systems (3<sup>rd</sup> ed.). New Delhi: Pearson Education.
3. W. Stallings, ( 2008). Operating Systems, Internals & Design Principles (5<sup>th</sup> ed.). Prentice Hall of India.

**Web Sites**

1. [www.cs.columbia.edu/~nieh/teaching/e6118\\_s00/](http://www.cs.columbia.edu/~nieh/teaching/e6118_s00/)
2. [www.clarkson.edu/~jnm/cs644](http://www.clarkson.edu/~jnm/cs644)
3. [pages.cs.wisc.edu/~remzi/Courses/736/Fall2002/](http://pages.cs.wisc.edu/~remzi/Courses/736/Fall2002/)

**18ITU313****RDBMS - PRACTICAL****Semester – III  
4H – 2C****Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100****End Semester Exam : 3 Hours****Course Objectives (CO)**

- To describe a sound introduction to the discipline of database management systems.
- To give a good formal foundation on the relational model of data and usage of Relational Algebra.
- To introduce the concepts of basic SQL as a universal Database language.
- To enhance knowledge to advanced SQL topics like embedded SQL, procedures connectivity through JDBC.
- To demonstrate the principles behind systematic database design approaches by covering conceptual design, logical design through normalization.
- To provide an overview of physical design of a database system, by discussing Database indexing techniques and storage techniques.

**Course Outcomes (COs)**

Upon completion of the course, students will be able to

1. Explain the features of database management systems and Relational database.
2. Design conceptual models of a database using ER modeling for real life applications and also construct queries in Relational Algebra.
3. Create and populate a RDBMS for a real life application, with constraints and keys, using SQL.
4. Retrieve any type of information from a data base by formulating complex queries in SQL.
5. Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.
6. Build indexing mechanisms for efficient retrieval of information from a database

**List of Programs**

1. Create a table with following fields:

Employee table:

Field name	Constraint	Type	Size
Employee_no	Primary key	Character	6
Employee_name		Character	30
Address		Character	25
Designation		Character	15
Dob		Date	
Gender	Check	Character	1
Doj		Date	
Salary		Number	10,2

**Queries:**

- a) Display name of the

employees whose salary is greater than “10,000”.

- b) Display the details of employees in ascending order according to Employee Code
- c) Display the details of employees earning the highest salary
- d) Display the names of employees who earn more than “Ravi”.

2. Create table named Student with following fields and insert the values:

Field name	field type	field size
Student Name	Character	15
Gender	Character	6
Roll No.	Character	10
Department Name	Character	15
Address	Character	25
Percentage of marks	Number	4,2

#### Queries:

- a) Calculate the average mark percentage of the students
- b) Display the names of the students whose percentage marks are greater than 80%
- c) Display the details of the students who got the highest percentage of marks
- d) Display the details of the students whose mark percentage between 50 and 70
- e) Display the details of the students whose mark percentage is greater than the mark percentage of Roll No=12CA01

3. Create a table with following fields:

Staff table:

Field name	Constraint	Type	Size
Staff_no	Primary key	Character	6
Staff_name		Character	30
Dob		Date	
Dept_code	Foreign key	Character	4
Designation		Character	15
Basic		Number	7,2

Department table:

Field name	constraint	Type	Size
Dept_code	Primary key	Character	4
Dept_name		Character	30

Execute the following queries:

1. To list the staff who joined 2 years back.
2. To list the staff in computer science dept.
3. To list the staff\_name and the dept\_name in which he/she works.
4. To list the maximum and minimum salary in each dept.
5. To list the dept along with the total amount spent on salary
6. To list the name of the employees who draw the salary more than the average salary.

4. Create a table with the following fields:



Book table:

Field name	Constraint	Type	Size
Access_no	Primary key	Character	6
Title		Character	30
Author		Character	30
Publisher		Character	30
Subject		Character	10
Price		Number	6,2

Execute the following queries:

1. The title of C and C++ books.
  2. The books written by a particular author.
  3. The books which costs between Rs.300/- and Rs.500/-
  4. The number of books available in each subject.
  5. The books in the decreasing order of the cost.
5. Create two tables course and batch with following fields  
 COURSE: coursecodeno number(5), course name varchar(20), syllabus varchar(20)  
 BATCH: bcode number(5), coursecode number(5), starting\_date date, duration number(3), coursefee number(10,2)

Perform the following queries

- Insert the details for course and batch tables with 10 records
- Show the description of the two tables
- Select all the fields from course & batch tables
- Select all the fields from course & batch tables where coursecode=10
- Select all the fields from batch table where starting date=march 10<sup>th</sup>
- Select batch code from batch table where net income>50000
- Select coursename, batch code & starting date from batch and course tables where course code of batch table and course code of course table are equal
- Select a syllabus from course wher coursecode=5

6. Create a table with the following fields:

Account table:

Field name	Constraint	Type	Size
Acc_no	Primary key	Number	4
Cust_name		Varchar2	30
Branch_name		Varchar2	30
Cust_city		Varchar2	30

Borrower table:

Field name	Constraint	Type	Size
Acc_no	Foreign key	Number	30
Branch_name		Varchar2	30
Amount		Number	8,2

Write queries to perform different types of join.

7. Write the PL/SQL program to find the factorial and fibonacci series of given numbers.

8.(i) Write the PL/SQL program to check whether the string is Palindrome.

- (ii) Write the PL/SQL program to reverse a number  
 (iii) Write the PL/SQL program to check whether the number is Armstrong

9. Write a PL/SQL block to create and handle user defined exception  
 clientmaster

Field name	Constraint	Type	Size
Client_id		Number	6
Client_name		Varchar2	30
Address		Varchar2	50
Phone		Number	10
Balance		Number	10,2

10. Create table with following fields:

Product table:

Field name	Constraint	Type	Size
Product_code	Primary key	Varchar2	7
Product_name		Varchar2	30
Price		Number	6,2
Quantity		Number	4

Vendor table:

Field name	Constraint	Type	Size
Vendor_name		Varchar2	30
Vendor address		Varchar2	30
Product_code	Foreign Key	Varchar2	7

Create a trigger to fire when the Record is deleted and inserted.

11. Write a PL/SQL trigger to update the records while deleting the one record in another table.

Voters\_master:

Field name	Constraint	Type	Size
Voterid	Primary key	Number	5
Name		Varchar2	30
Ward_no	Primary Key	Number	4
Dob		Date	
Address		Varchar2	150

New\_list

Field name	Constraint	Type	Size
Voterid		Number	5
Ward_no		Number	4
Name		Varchar2	30
Description		Character	50

12. Create a table to store the salary details of the employees in a company. Declare the cursor id to contain empno, employee name and net salary. Use cursor to update the employee details.

Salary:

Field name	Constraint	Type	Size
Emp_no	Primary key	Number	4
Emp_name		Varchar2	30
Designation		Varchar2	25
Dept		Varchar2	30
Basic		Number	5

13. Create a table stock contains the itemcode varchar2(10), itemname varchar2(50), current\_stock number(5), date\_of\_last\_purchase date. Write a stored procedure to seek for an item using itemcode and delete it, if the date of last purchase is before 1 year from the current date. If not, update the current stock.

14. Create a table to contain phone\_number, user\_name, address. Write a function to search for address using phone\_number.

### Suggested Readings

1. Bipin C. Desai.(2013). An Introduction to Database Systems, New Delhi: Galgotia Publications.
2. Rajiv chopra (2013). Database Management systems (3<sup>rd</sup> ed.). S.Chand publications.
3. Steven Feurstein, Bill Pribyl (2014). Oracle PL/SQL Programming (6<sup>th</sup> ed.). O ‘ Reilly Media.
4. Shio Kumar Singh (2011). Database Management Systems – Concepts, design and Applications (2<sup>nd</sup> ed.). New Delhi: Pearson Education.
5. Ivan Byross (2010). SQL, PL/SQL the Programming Language of Oracle Paperback. BPB Publications.
6. Rajeeb C. Chatterjee (2012). Learning Oracle SQL and PL/SQL: A simplified Guide. Prentice Hall of India.

### Web Sites

1. <http://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm>
2. [www.databasedir.com](http://www.databasedir.com)
3. <http://plsqli-tutorial.com/>

**Semester – III**

**18ITU314A**

**ANDROID PROGRAMMING - PRACTICAL**

**3H – 1C**

**Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Int : 40 Ext : 60****Total: 100****End Semester Exam : 3 Hours****Course Objectives (CO)**

- To introduce Android platform and its architecture.
- To learn activity creation and Android UI designing.
- To be familiarized with Intent, Broadcast receivers and Internet services.
- To work with SQLite Database and content providers.
- To integrate multimedia, camera and Location based services in Android Application.
- To explore Mobile security issues.

**Course Outcomes (COs)**

Upon completion of this course, the students will able to

1. Describe Android platform, Architecture and features.
2. Design User Interface and develop activity for Android App.
3. Use Internet, Broadcast receivers and Internet services in Android App.
4. Design and implement Database Application and Content providers.
5. Use multimedia, camera and Location based services in Android App.
6. Discuss various security issues in Android platform

**List of Programs**

1. Create —Hello World application. That will display —Hello World in the middle of the screen in the emulator. Also display —Hello World in the middle of the screen in the Android Phone.
2. Create an application with login module. (Check username and password).
3. Create spinner with strings taken from resource folder (res >> value folder) and on changing the spinner value, Image will change.
4. Create a menu with 5 options and selected option should appear in text box.
5. Create a list of all courses in your college and on selecting a particular course teacher-in-charge of that course should appear at the bottom of the screen.
6. Create an application with three option buttons, on selecting a button colour of the screen will change.
7. Create and Login application as above. On successful login, pop up the message.
8. Create an application to Create, Insert, update, Delete and retrieve operation on the database.

**Suggested Readings**

1. James, C. Sheusi.( 2013). Android application development for java rogrammers. Cengage Learning.

**Web Sites**

1. <http://www.developer.android.com>
2. <http://developer.android.com/about/versions/index.html>
3. <http://developer.android.com/training/basics/firstapp/index.html>
4. <http://docs.oracle.com/javase/tutorial/index.htm>(Available in the form of free downloadable ebooks also).
5. <http://developer.android.com/guide/components/activities.html>
6. <http://developer.android.com/guide/components/fundamentals.html>
7. <http://developer.android.com/guide/components/intents-filters.html>.
8. <http://developer.android.com/training/multiscreen/screensizes.html>
9. <http://developer.android.com/guide/topics/ui/controls.html>
10. <http://developer.android.com/guide/topics/ui/declaring-layout.html>
11. <http://developer.android.com/training/basics/data-storage/databases.html>

**Semester – III****18ITU314B****PROGRAMMING IN PYTHON- PRACTICAL****3H – 1C**

**Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Internal: 40 External: 60 Total: 100**  
**End Semester Exam: 3 Hours**

**Course Objectives (CO)**

- To know the basics of algorithmic problem solving
- To read and write simple Python programs.
- To develop Python programs with conditionals and loops.
- To define Python functions and call them.
- To use Python data structures — lists, tuples, dictionaries.
- To do input/output with files in Python.

**Course Outcomes (COs)**

Upon completion of the course, students will be able to

1. Develop algorithmic solutions to simple computational problems
2. Read, write, execute by hand simple Python programs.
3. Structure simple Python programs for solving problems.
4. Decompose a Python program into functions.
5. Represent compound data using Python lists, tuples, dictionaries.
6. Read and write data from/to files in Python Programs.

**List of Programs**

1. Compute the GCD of two numbers.
2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria :
  - Grade A: Percentage  $\geq 80$
  - Grade B: Percentage  $\geq 70$  and  $< 80$
  - Grade C: Percentage  $\geq 60$  and  $< 70$
  - Grade D: Percentage  $\geq 40$  and  $< 60$
  - Grade E: Percentage  $< 40$
3. WAP to display the first n terms of Fibonacci series.
4. WAP to find factorial of the given number.
5. Find the square root of a number (Newton's method)
6. WAP to find the Exponentiation (power of a number)
7. Find the maximum of a list of numbers

8. WAP to perform Linear search
9. WAP to perform Binary search
10. WAP to perform Selection sort
11. WAP to find first n prime numbers
12. WAP to calculate the Multiply matrices
13. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's choice.
14. Write a program using file operations.

### Suggested Readings

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist'', 2nd edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016 (<http://greenteapress.com/wp/thinkpython/>)
2. Guido van Rossum and Fred L. Drake Jr, —An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.
3. John V Guttag, —Introduction to Computation and Programming Using Python'', Revised and expanded Edition, MIT Press , 2013
4. Robert Sedgewick, Kevin Wayne, Robert Dondero, —Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016.
5. Timothy A. Budd, —Exploring Python, Mc-Graw Hill Education (India) Private Ltd., 2015.
6. Kenneth A. Lambert, —Fundamentals of Python: First Programs, CENGAGE Learning, 2012.
7. Charles Dierbach, —Introduction to Computer Science using Python: A Computational ProblemSolving Focus, Wiley India Edition, 2013.
8. Paul Gries, Jennifer Campbell and Jason Montojo, —Practical Programming: An Introduction to Computer Science using Python 3, Second edition, Pragmatic Programmers, LLC, 2013

**18ITU401 DATA COMMUNICATION AND NETWORKS****4H – 4C**

**Instruction Hours / week: L: 4 T: 0 P: 0 Marks: Int : 40 Ext : 60 Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives**

- To study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model.
- To acquire knowledge of Application layer and Presentation layer paradigms and protocols.
- To study Session layer design issues, Transport layer services, and protocols.
- To gain core knowledge of Network layer routing protocols and IP addressing.
- To study data link layer concepts, design issues, and protocols.
- To read the fundamentals and basics of Physical layer, and will apply them in real time applications.

**Course Outcomes (COs)**

1. Describe the functions of each layer in OSI and TCP/IP model.
2. Explain the functions of Application layer and Presentation layer paradigms and Protocols.
3. Describe the Session layer design issues and Transport layer services.
4. Classify the routing protocols and analyze how to assign the IP addresses for the given network.
5. Describe the functions of data link layer and explain the protocols.
6. Explain the types of transmission media with real time applications

**Unit I**

**Introduction to Data Communication:** Network, Protocols & standards and standards organizations - Line Configuration; layered network architecture; overview of OSI reference model; overview of TCP/IP protocol suite. **Data Communication Fundamentals and Techniques:** Analog and digital signal; data-ratelimits; digital to digital line encoding schemes; pulse code modulation; parallel and serial transmission;

**Unit II**

( cont..)digital to analog modulation-; multiplexing techniques- FDM, TDM; transmission media.

**Networks Switching Techniques and Access mechanisms:** Circuit switching; packet switching - connectionless datagram switching, connection-oriented virtual circuit switching; dial-up modems; digital subscriber line; cable TV for data transfer.

**Unit III**

**Data Link Layer Functions and Protocol:** Error detection and error correction techniques; data-link control- framing and flow control; error recovery protocols- stop and wait ARQ, go-back-n ARQ; Point to Point Protocol on Internet.

**Unit IV**



**Multiple Access Protocol and Networks:** CSMA/CD protocols; Ethernet LANs; connecting LAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways; **Networks Layer Functions and Protocols:** Routing; routing algorithms; network layer protocol of Internet- IP protocol, Internet control protocols.

## Unit V

**Transport Layer Functions and Protocols:** Transport services- error and flow control, Connection establishment and release- three way handshake; **Overview of Application layer protocol:** Overview of DNS protocol; overview of WWW & HTTP protocol.

## Suggested Readings

1. Forouzan, B. A.( 2007). Data Communications and Networking(4<sup>th</sup> ed.). New Delhi: THM.
2. Tanenbaum, A. S. (2002). Computer Networks (4<sup>th</sup> ed.). New Delhi: PHI.

## Web Sites

1. [en.wikipedia.org/wiki/Internet\\_protocol\\_suite](http://en.wikipedia.org/wiki/Internet_protocol_suite)
2. [http://docwiki.cisco.com/wiki/Introduction\\_to\\_WAN\\_Technologies](http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies)
3. [www.yale.edu/pclt/COMM/TCPIP.HTM](http://www.yale.edu/pclt/COMM/TCPIP.HTM)
4. [www.w3schools.com/tcpip/default.asp](http://www.w3schools.com/tcpip/default.asp)

**18ITU402****SOFTWARE ENGINEERING****Semester – IV  
4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100****End Semester Exam : 3 Hours****Course Objectives (CO)**

- To understand the nature of software development and software life cycle process models, agile software development, SCRUM and other agile practices.
- To explain methods of capturing, specifying, visualizing and analyzing software requirements.
- To understand concepts and principles of software design and user-centric approach and principles of effective user interfaces. 4. To know basics of testing and understanding concept of software quality assurance and software configuration management process.
- To understand the need of project management and project management life cycle.
- To understand project scheduling concept and risk management associated to various type of projects.
- Implement a given software design using sound development practices.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Apply their knowledge of mathematics, sciences, and computer science to the modeling, analysis, and measurement of software artifacts.
2. Work effectively as leader/member of a development team to deliver quality software artifacts.
3. Analyze, specify and document software requirements for a software system.
4. Verify, validate, assess and assure the quality of software artifacts.
5. Design, select and apply the most appropriate software engineering process for a given project, plan for a software project, identify its Course Objectives and risks, and estimate its cost and time.
6. Express and understand the importance of negotiation, effective work habits, leadership, and good communication with stakeholders, in written and oral forms, in a typical software development environment.

**Unit I**

**Introduction:** The Evolving Role of Software, Software Characteristics, Changing Nature of Software, Software Engineering as a Layered Technology, Software Process Framework, Framework and Umbrella Activities, Process Models, Capability Maturity Model Integration (CMMI).

**Unit II**

**Requirement Analysis:** Initiating Requirement Engineering Process- Requirement Analysis and Modeling Techniques- Flow Oriented Modeling- Need for SRS- Characteristics and Components of SRS- Software Project Management: Estimation in Project Planning Process, Project Scheduling.

**Unit III**

**Risk Management:** Software Risks, Risk Identification Risk Projection and Risk Refinement, RMMM plan, **Quality Management-** Quality Concepts, Software Quality Assurance, Software Reviews, Metrics for Process and Projects

**Unit IV**

**Design Engineering:** Design Concepts, Architectural Design Elements, Software Architecture, Data Design at the Architectural Level and Component Level, Mapping of Data Flow into Software Architecture, Modeling Component Level Design

**Unit V**

**Testing Strategies & Tactics:** Software Testing Fundamentals, Strategic Approach to Software Testing, Test Strategies for Conventional Software, Validation Testing, System testing Black-Box Testing, White-Box Testing and their type, Basis Path Testing

**Suggested Readings**

- 1.R.S. Pressman, (2009). Software Engineering: A Practitioner's Approach (7<sup>th</sup> ed.). McGraw-Hill.
- 2.P.Jalote (2008). An Integrated Approach to Software Engineering (2<sup>nd</sup> ed.). New Age International Publishers.
- 3.K.K. Aggarwal and Y.Singh (2008). Software Engineering ( 2<sup>nd</sup> ed.). New Age International Publishers.
- 4.Sommerville (2006). Software Engineering (8<sup>th</sup> ed.). Addison Wesley.
- 5.D.Bell (2005). Software Engineering for Students (4<sup>th</sup> ed.) Addison-Wesley.
- 6.R.Mall (2004). Fundamentals of Software Engineering (2<sup>nd</sup> ed.). Prentice-Hall of India.

**Web Sites**

1. [http://en.wikipedia.org/wiki/Software\\_engineering](http://en.wikipedia.org/wiki/Software_engineering)
2. <http://www.onesmartclick.com/engineering/software-engineering.html>
3. [http://www.CC.gatech.edu/classes/AY2000/cs3802\\_fall/](http://www.CC.gatech.edu/classes/AY2000/cs3802_fall/)

**18ITU403****PROGRAMMING IN PERL****4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100****End Semester Exam : 3 Hours****Course Objectives (CO)**

1. To experience learning a programming language "on your own" as is commonly the case in industry.
2. To understand the syntax and semantics of the Perl language and their similarity and differences from Java
3. To understand how to develop and implement various types of programs in the Perl language
4. To understand various forms of data representation and structures supported by the Perl language
5. To understand the appropriate applications of the Perl language
6. To recognize similarities and common characteristics of programming languages.

**Course Outcome (COs)**

After the completion of this course, a successful student will be able to:

1. Focus on new type of programming methods
2. Analyze the ease of the language with other languages
3. Familiar with implementation of CGI
4. Interpret the mathematical results in physical and other forms.
5. Identify, formulate and solve the Linear Differential Equations.
6. Classify and solve the contour integration of complex functions.

**Unit I**

**Introduction:** Programming Languages – Why Perl a great Language? – ASCII & Unicode – Types of Data – Operators – Variables – Interpolation – List, Arrays & Hashes.

**Unit- II**

**Control Structures, Regular Expression and file:** Loops and Decisions: Deciding if – Skip few – Looping While – Controlling loop flow – Regular Expression: Patterns – Working with regexp – Advanced topics. File and Data: File Handles – Writing to files – opening pipes – directories

**Unit- III**

**References and Subroutine:** Life Cycle – Reference with complex Data Structure. Subroutine: Understanding subroutine – Subroutine for calculations – Course Objectives in functions – Passing Complex Parameters – References to subroutine

**Unit- IV**

**Modules & CGI:** Modules: Types – Package hierarchy – Exporters – Standard modules. Object oriented Perl: Rolling your Own – Inheritance: Introduction to CGI: Writing CGI Scripts – Writing Interactive CGI Scripts – Debugging CGI Scripts

**Unit- V**

**Networking and IPC:** Networking: Obtaining Network Information – The Socket module – Socket Communication – Using IO Socket. IPC: Process – Signals – Pipes – Executing Additional Process

**Suggested Readings**

1. Simon Cozen & Peter Wainwright (2000). Beginning Perl (1st Edition.). Wrox Press, USA
2. Martin C Brown (2001). Perl: The complete Reference (2<sup>nd</sup> Edition) McGraw Hill Education (Indian) Pvt.ltd, Noida.
3. Tom Christiansen, Brain D foy, Larry Wall (2012). Perl Programming. OReilly Publications
4. Randal L Schwartz, Brain D Foy, Tom Phoneix(2017) . Learning Perl. OReilly Publications
5. Brain D Foy (2012), Intermediate Perl, OReilly Publications

**Web Sites**

1. [www.Perlmaven.com](http://www.Perlmaven.com)
2. [www.javapoint.com/perl-tutorial](http://www.javapoint.com/perl-tutorial)
3. [www.perltutorial.org](http://www.perltutorial.org)
4. [www.tutorialpoint.com/perl](http://www.tutorialpoint.com/perl)
5. [www.learn-perl.org](http://www.learn-perl.org)

**18ITU404A****SCRIPTING LANGUAGE****Semester – IV  
3H – 3C****Instruction Hours / week: L: 3 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100  
End Semester Exam : 3 Hours****Course Objectives (CO)**

- To classify the various Scripting Languages
- To learn client and server side scripting languages ( Java script and AJAX, JSP)
- To create simple Web pages and provide client side validation.
- To create dynamic web pages using server side scripting
- To master the theory behind scripting and its relationship to classic programming
- To gain some fluency programming in JavaScript, AJAX, and related languages, to design and implement one's own scripting language.

**Course Outcomes (COs)**

At the end of the course, the student should be able to:

1. Define the CSS with its types and use them to provide the styles to the web pages at various levels.
2. Develop the modern web pages using the HTML and CSS features with different layouts as per need of applications.
3. Use the JavaScript to develop the dynamic web pages.
4. Use server side scripting with JSP to generate the web pages dynamically.
5. Gain knowledge of client side scripting, validation of forms and AJAX programming.  
Create applications by using the concepts like JSP and Servlet

**Unit I**

**Introduction To VB script:** Introduction- Embedding VBScript Code in an HTML Document  
Comments-Variables- Operators-Procedures- Conditional Statements- Looping Constructs -  
Objects and VBScript – Cookies.

**Unit II**

**Introduction To Java Script :** JavaScript- Introduction, simple programming, Obtaining User Input with prompt Dialogs, Operators (arithmetic, Decision making, assignment, logical, increment and decrement. Functions - program modules in JavaScript, programmer defined

functions, function definition, Random-number generator, scope rules, global functions, recursion.

### Unit III

**Functions, Arrays And Objects :** JavaScript: Arrays, Objects - Math Object, String Object, Date Object, Boolean & Number Object, document and window Objects. Handling event using java script

### Unit IV

**Client Side Technologies : AJAX**– Evolution of AJAX – AJAX Framework – Web applications with AJAX – AJAX with PHP – AJAX with Databases- Ajax Client Server Architecture-XML Http Request Object-Call Back Methods.

### Unit V

**Server Side Scripting- JSP :** Servlet Overview – Life cycle of a Servlet – Handling HTTP request and response – Using Cookies – Session tracking – Java Server Pages – Anatomy of JSP – Implicit JSP Objects – JDBC – Java Beans – Advantages – Enterprise Java Beans – EJB Architecture – Types of Beans – EJB Transactions

### Suggested Readings

1. Jeffrey C. Jackson, "Web Technologies--A Computer Science Perspective", Pearson Education, 2006.
2. Deitel, Deitel, Goldberg, "Internet & World Wide Web How to Program", Third Edition, Pearson Education, 2006.
3. Bryan Basham, Kathy Siegra, Bert Bates, "Head First Servlets and JSP", Second Edition
4. Uttam K Roy, "Web Technologies", Oxford University Press, 2011.
5. Robert. W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson Education, 2007 .
6. Marty Hall and Larry Brown, "Core Web Programming" Second Edition, Volume I and II, Pearson Education, 2001.

**Semester – IV****18ITU404B****XML PROGRAMMING****3H – 3C****Instruction Hours / week: L: 3 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100****End Semester Exam : 3 Hours****Course Objectives (CO)**

- To expose the students to the XML family of technologies, and the latest W3C and WS-I XML standards.
- To understand the various applications of XML in the areas of information representation, Presentation Oriented Publishing, Message Oriented computing, and Application Configuration.
- To expose the students to the combined use of XML and Java technologies
- To support the development of modern applications targeted to the evolving spectrum of distributed and decentralized enterprise platforms.
- To expose the students to the advanced XML-enabled capabilities of the Java 2 development environment for Enterprise Applications.
- To demonstrate the application of XML in distributed communications enabling, enterprise systems assurance, web enabling, application enabling, and enterprise data enabling.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Create a new webpage
2. Understand the fundamental features of web applications.
3. Understand the objects and components needed for a web designing.
4. Understand the current industry support for XML technologies.
5. Sharpen the students' practical development skills via focused assignments and projects.
6. Understand what is XML and how to parse and use XML Data

**Unit I**

**Introduction:** Understanding Mark-up Languages, Introduction to XML and its Goals.

**Unit II**

**XML Basics:** XML Structure and Syntax, Document classes and Rules.

**Unit III**

**Other XML Concepts:** Scripting XML

**Unit IV**

**Other XML Concepts:** XML as Data, Linking with XML

**Unit V**

**XML with Style:** XSL –Style Sheet Basics, XSL basics, XSL style sheets.



**Suggested Readings**

1. William, J. Pardi. XML in action web technology.
2. Michael, J. Young. Step by Step XML.

**Web Sites**

1. <http://java.sun.com/features/2001/02/xmlj2ee.p.html>.
2. <http://www.w3.org/XML/1999/xml-in-10-points.html.en>

**18ITU411 DATA COMMUNICATION AND NETWORKS  
- PRACTICAL****Semester – IV  
4H – 2C**

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100  
End Semester Exam : 3 Hours****Course Objectives (CO)**

- To study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model.
- To acquire knowledge of Application layer and Presentation layer paradigms and protocols.
- To study Session layer design issues, Transport layer services, and protocols.
- To gain core knowledge of Network layer routing protocols and IP addressing.
- To study data link layer concepts, design issues, and protocols.
- To read the fundamentals and basics of Physical layer, and will apply them in real time applications.

**Course Outcomes (COs)**

1. Describe the functions of each layer in OSI and TCP/IP model.
2. Explain the functions of Application layer and Presentation layer paradigms and Protocols.
3. Describe the Session layer design issues and Transport layer services.
4. Classify the routing protocols and analyze how to assign the IP addresses for the given network.
5. Describe the functions of data link layer and explain the protocols.
6. Explain the types of transmission media with real time applications

**List of Programs**

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate and implement stop and wait protocol for noisy channel.
3. Simulate and implement go back n sliding window protocol.
4. Simulate and implement selective repeat sliding window protocol.
5. Simulate and implement distance vector routing algorithm
6. Simulate and implement Dijkstra algorithm for shortest path routing.

**Suggested Readings**

1. Forouzan, B. A. (2007). Data Communications and Networking (4<sup>th</sup> ed.). New Delhi: THM.
2. Tanenbaum, A. S. (2002). Computer Networks (4<sup>th</sup> ed.). New Delhi: PHI.

**Web Sites**

1. [en.wikipedia.org/wiki/Internet\\_protocol\\_suite](http://en.wikipedia.org/wiki/Internet_protocol_suite)
2. [http://docwiki.cisco.com/wiki/Introduction\\_to\\_WAN\\_Technologies](http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies)
3. [www.yale.edu/pclt/COMM/TCPIP.HTM](http://www.yale.edu/pclt/COMM/TCPIP.HTM)
4. [www.w3schools.com/tcpip/default.asp](http://www.w3schools.com/tcpip/default.asp)

**Semester – IV****18ITU412****SOFTWARE ENGINEERING - PRACTICAL****4H – 2C****Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100****End Semester Exam : 3 Hours****Course Objectives (CO)**

- To understand the nature of software development and software life cycle process models, agile software development, SCRUM and other agile practices.
- To explain methods of capturing, specifying, visualizing and analyzing software requirements.
- To understand concepts and principles of software design and user-centric approach and principles of effective user interfaces. 4. To know basics of testing and understanding concept of software quality assurance and software configuration management process.
- To understand the need of project management and project management life cycle.
- To understand project scheduling concept and risk management associated to various type of projects.
- To analyze, specify and document software requirements for a software system.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Apply their knowledge of mathematics, sciences, and computer science to the modeling, analysis, and measurement of software artifacts.
2. Work effectively as leader/member of a development team to deliver quality software artifacts.
3. Implement a given software design using sound development practices.
4. Verify, validate, assess and assure the quality of software artifacts.
5. Design, select and apply the most appropriate software engineering process for a given project, plan for a software project, identify its Course Objectives and risks, and estimate its cost and time.
6. Express and understand the importance of negotiation, effective work habits, leadership, and good communication with stakeholders, in written and oral forms, in a typical software development environment.

**List of Programs**

S. No	Practical Title
1.	<ul style="list-style-type: none"> <li>• Problem Statement</li> <li>• Process Model</li> </ul>
2.	Requirement Analysis: <ul style="list-style-type: none"> <li>• Creating a Data Flow</li> <li>• Data Dictionary, Use Cases</li> </ul>
3.	Project Management:

	<ul style="list-style-type: none"> <li>• Computing FP</li> <li>• Effort</li> <li>• Schedule, Risk Table, Timeline chart</li> </ul>
<b>4.</b>	Design Engineering: <ul style="list-style-type: none"> <li>• Architectural Design</li> <li>• Data Design, Component Level Design</li> </ul>
<b>5.</b>	Testing: <ul style="list-style-type: none"> <li>• Basis Path Testing</li> </ul>

### Sample Projects:

1. **Criminal Record Management:** Implement a criminal record management system for jailers, police officers and CBI officers
2. **DTC Route Information:** Online information about the bus routes and their frequency and fares
3. **Car Pooling:** To maintain a web based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.
4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation system
8. Automatic Internal Assessment System
9. Parking Allocation System
10. Wholesale Management System

### Suggested Readings

- 1.R.S. Pressman, (2009). Software Engineering: A Practitioner's Approach (7<sup>th</sup> ed.). McGraw-Hill.
- 2.P.Jalote (2008). An Integrated Approach to Software Engineering (2<sup>nd</sup> ed.). New Age International Publishers.
- 3.K.K. Aggarwal and Y.Singh (2008). Software Engineering ( 2<sup>nd</sup> ed.). New Age International Publishers.
- 4.Sommerville (2006). Software Engineering (8<sup>th</sup> ed.). Addison Wesley.
- 5.D.Bell (2005). Software Engineering for Students (4<sup>th</sup> ed.) Addison-Wesley.
- 6.R.Mall (2004). Fundamentals of Software Engineering (2<sup>nd</sup> ed.). Prentice-Hall of India.

### Web Sites

1. [http://en.wikipedia.org/wiki/Software\\_engineering](http://en.wikipedia.org/wiki/Software_engineering)
2. <http://www.onesmartclick.com/engineering/software-engineering.html>
3. [http://www.CC.gatech.edu/classes/AY2000/cs3802\\_fall/](http://www.CC.gatech.edu/classes/AY2000/cs3802_fall/)

**Semester – IV****18ITU413****PROGRAMMING IN PERL - PRACTICAL****4H – 2C**

---

**Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Int : 40 Ext : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To experience learning a programming language "on your own" as is commonly the case in industry.
- To understand the syntax and semantics of the Perl language and their similarity and differences from Java
- To understand how to develop and implement various types of programs in the Perl language
- To understand various forms of data representation and structures supported by the Perl language
- To understand the appropriate applications of the Perl language
- To recognize similarities and common characteristics of programming languages.

**Course Outcome (COs)**

After the completion of this course, a successful student will be able to:

1. Focus on new type of programming methods
2. Analyse the ease of the language with other languages
3. Familiar with implementation of CGI
4. Interpret the mathematical results in physical and other forms.
5. Identify, formulate and solve the Linear Differential Equations.
6. Classify and solve the contour integration of complex functions.

**List of Programs**

1. Write a program that converts
  - a) Hexadecimal to Decimal
  - b) Octal to Decimal
2. Write a program which gets the decimal value less than 256 and convert it to binary using only bitwise operator.
3. Write a program which stores the phone number in a hash and retrieve them by using the person's name.

4. Write a program that displays the prime number between 2 to the given number 'n'.
5. Write a program that
  - a) Counts the number of 'a' in a given paragraph
  - b) Replace multiple white space with single white space
6. Write a program that sorts the given words in alphabetical order using regular expression. The list of words must be separated with new line characters and the program must use bubble sort.
7. Write a program that can search a specified string within all the files in a given directory.
8. Write a program that sends an array of integers to a subroutine that stores the odd numbers and even numbers in different arrays and returns the sum of even numbers only.
9. Write a program that collects the integers and sends only the even number as an array to a subroutine that displays the biggest of the received even numbers.
10. Write a program that displays the basic network information of a system
11. Write a program that executes additional process during IPC.
12. Write a program that gets a set of words from the user and display the words by avoiding duplicate words.

### Suggested Readings

1. Simon Cozen & Peter Wainwright (2000). Beginning Perl (1st Edition.). Wrox Press, USA
2. Martin C Brown (2001). Perl: The complete Reference (2<sup>nd</sup> Edition) McGraw Hill Education (Indian) Pvt.ltd, Noida.
3. Tom Christiansen, Brain D foy, Larry Wall (2012). Perl Programming. OReilly Publications
4. Randal L Schwartz, Brain D Foy, Tom Phoneix(2017) . Learning Perl. OReilly Publications
5. Brain D Foy (2012), Intermediate Perl, OReilly Publications

### Web Sites

1. [www.Perlmaven.com](http://www.Perlmaven.com)
2. [www.javapoint.com/perl-tutorial](http://www.javapoint.com/perl-tutorial)
3. [www.perltutorial.org](http://www.perltutorial.org)
4. [www.tutorialpoint.com/perl](http://www.tutorialpoint.com/perl)
5. [www.learn-perl.org](http://www.learn-perl.org)

**Semester – IV****18ITU414A****SCRIPTING LANGUAGE - PRACTICAL****3H – 1C**

---

**Instruction Hours / week: L: 0 T: 0 P: 3    Marks: Int : 40 Ext : 60    Total: 100**  
**End Semester Exam : 3 Hours****Course Objectives (CO)**

- To classify the various Scripting Languages
- To learn client and server side scripting languages ( Java script and AJAX, JSP)
- To create simple Web pages and provide client side validation.
- To create dynamic web pages using server side scripting
- To master the theory behind scripting and its relationship to classic programming
- To gain some fluency programming in JavaScript, AJAX, and related languages, to design and implement one's own scripting language.

**Course Outcomes (COs)**

At the end of the course, the student should be able to:

1. Define the CSS with its types and use them to provide the styles to the web pages at various levels.
2. Develop the modern web pages using the HTML and CSS features with different layouts as per need of applications.
3. Use the JavaScript to develop the dynamic web pages.
4. Use server side scripting with JSP to generate the web pages dynamically.
5. Gain knowledge of client side scripting, validation of forms and AJAX programming.
6. Create applications by using the concepts like JSP and Servlet.

**List of Programs**

1. Create Application form using various text formats.
2. Create UNIVERSITY website using HTML tags.
3. Create a table using HTML.
4. Display your information using form controls.
5. Create style sheets with the style elements.



6. Create calculator format using java script.
7. Create an array of 10 numbers and sort them using javascript.
8. String manipulation using string object.
9. Add a simple script using Click event.
10. Create Employee details using schemas.
11. Create our department details using CSS.
12. Create Payroll system using XSL.
13. Changing image using mouseover event.
14. Create a website for a newspaper.
15. Design and apply your application form for course enrolment using Javascript.

### **Suggested Readings**

1. Jeffrey C. Jackson, "Web Technologies--A Computer Science Perspective", Pearson Education, 2006.
2. Deitel, Deitel, Goldberg, "Internet & World Wide Web How to Program", Third Edition, Pearson Education, 2006.
3. Bryan Basham, Kathy Siegra, Bert Bates, "Head First Servlets and JSP", Second Edition
4. Uttam K Roy, "Web Technologies", Oxford University Press, 2011.
5. Robert. W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson Education, 2007 .
6. Marty Hall and Larry Brown, "Core Web Programming" Second Edition, Volume I and II, Pearson Education, 2001.

**Semester – IV****18ITU414B****XML PROGRAMMING - PRACTICAL****3H – 1C****Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives (CO)**

- To expose the students to the XML family of technologies, and the latest W3C and WS-I XML standards.
- To understand the various applications of XML in the areas of information representation, Presentation Oriented Publishing, Message Oriented computing, and Application Configuration.
- To expose the students to the combined use of XML and Java technologies
- To support the development of modern applications targeted to the evolving spectrum of distributed and decentralized enterprise platforms.
- To expose the students to the advanced XML-enabled capabilities of the Java 2 development environment for Enterprise Applications.
- To demonstrate the application of XML in distributed communications enabling, enterprise systems assurance, web enabling, application enabling, and enterprise data enabling.

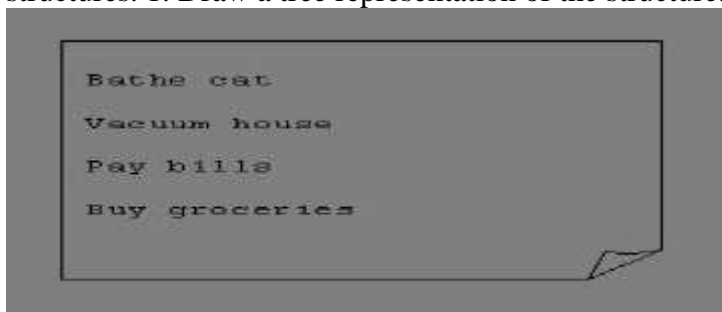
**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Create a new webpage
2. Understand the fundamental features of web applications.
3. Understand the objects and components needed for a web designing.
4. Understand the current industry support for XML technologies.
5. Sharpen the students' practical development skills via focused assignments and projects.
6. Understand what is XML and how to parse and use XML Data

**List of Programs****1. Information Structure**

In this exercise, student will practice identifying the structure of an information object. For the sample document provided below: Label the information structures you see, including containing structures. 1. Draw a tree representation of the structure.

**2. Deconstructing an XML Document**

In this exercise, student will practice identifying the explicit structure within an XML document. In a sense, this is the reverse of what you did in Exercise #1. For the sample XML markup below, create a document-like representation (or a simple drawing) for the content contained within the XML tags:

```
<book>
<coverInfo>
<title>The XML Handbook</title>
<author>Charles F. Goldfarb</author>
<author>Paul Prescod</author>
<edition>Second</edition>
<description>The definitive XML resource: applications, products, and technologies. Revised
and expanded—over 600 new pages. </description>
</coverInfo> </book>
```

### 3. Creating XML Markup

In this exercise, create some XML markup based on the tree representation from Exercise #1 above, and the content from the original sample document.

### 4. Well-Formedness

This exercise checks your understanding of the constraints for well-formedness. Are the following document instances well-formed? Explain any NO answers.

```
<list><title>The first list</title><item>An item</list>
<item>An item</item><item>Another item</item>
<para>Bathing a cat is a <emph>relatively</emph> easy task as long as the cat is
willing.</para>
<bibl><title>How to Bathe a Cat<author></title>Merlin Bauer<author></bibl>
```

### 5. Well Formedness

This exercise is a bit more challenging than the previous example. Here is a fragment of an XML document instance. Identify all the places where it fails to match the constraints for well-formedness.

```
<PROCEDURE><TITLE>How to Bathe a Cat</TITLE>
<OVERVIEW> This procedure tells you how to bathe a cat. <WARNING></OVERVIEW>Cats
don't like to take baths. You could get hurt doing this. Be sure to obtain all the required
protective gear before you start. </WARNING><EQUIPEMENT><ITEM>Hockey Mask
<ITEM>Padded Full-body Kevlar Armor</ITEM><ITEM>Tub full of warm
water</ITEM><ITEM>Towels </ITEM><ITEM>First Aid kit</ITEM><ITEM>Cat
Shampoo</ITEM> <EQUIPMENT><INSTRUCTIONS> <STEP> Locate the cat, who by now
is hiding under the bed.</STEP><STEP>Place the cat in the tub of water.</STEP>
<ITEM>Using the First Aid kit, repair the damage to your head and arms.</STEP>
<STEP>Place the cat back in the tub and hold it down.</STEP> <STEP>Wash it really fast, then
make an effort to dry it with the towels.</STEP> <STEP>Decide not to do this again. </STEP>
</INSTRUCTIONS>
```

### **Suggested Readings**

1. William, J. Pardi. XML in action web technology.
2. Michael, J. Young. Step by Step XML.

### **Web Sites**

1. <http://java.sun.com/features/2001/02/xmlj2ee.p.html>.
2. <http://www.w3.org/XML/1999/xml-in-10-points.html.en>

**18ITU501A****ARTIFICIAL INTELLIGENCE****Semester – V**  
**4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours****Course Objectives (CO)**

- To understand the various characteristics of Intelligent agents
- To learn about the different search strategies in AI
- To learn to represent knowledge in solving AI problems
- To understand the different ways of designing software agents
- To know about the various applications of AI.
- Understand the various searching techniques, constraint satisfaction problem and example problems- game playing techniques.

**Course Outcomes (COs)**

At the end of the course, the student should be able to:

1. Identify problems that are amenable to solution by AI methods.
2. Identify appropriate AI methods to solve a given problem.
3. Formalize a given problem in the language/framework of different AI methods.
4. Implement basic AI algorithms.
5. Design and carry out an empirical evaluation of different algorithms on a problem formalization, and state the conclusions that the evaluation supports.
6. Ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems

**Unit I - INTRODUCTION TO AI AND PRODUCTION SYSTEMS**

Introduction to AI-Problem formulation, Problem Definition -Production systems, Control strategies, Search strategies. Problem characteristics, Production system characteristics - Specialized production system- Problem solving methods – Problem graphs, Matching, Indexing and Heuristic functions -Hill Climbing-Depth first and Breadth first, Constraints satisfaction – Related algorithms, Measure of performance and analysis of search algorithms.

**Unit II - REPRESENTATION OF KNOWLEDGE**

Game playing – Knowledge representation, Knowledge representation using Predicate logic, Introduction to predicate calculus, Resolution, Use of predicate calculus, Knowledge representation using other logic-Structured representation of knowledge.

**Unit III - KNOWLEDGE INFERENCE**

Knowledge representation -Production based system, Frame based system. Inference – Backward chaining, Forward chaining, Rule value approach, Fuzzy reasoning – Certainty factors, Bayesian Theory-Bayesian Network-Dempster – Shafer theory.

**Unit IV- PLANNING AND MACHINE LEARNING**

Basic plan generation systems – Strips -Advanced plan generation systems – K strips -Strategic explanations -Why, Why not and how explanations. Learning- Machine learning, adaptive

Learning.

### **Unit V - EXPERT SYSTEMS**

Expert systems – Architecture of expert systems, Roles of expert systems – Knowledge Acquisition – Meta knowledge, Heuristics. Typical expert systems – MYCIN, DART, XOON, Expert systems shells.

#### **Suggested Readings**

1. Kevin Night and Elaine Rich, Nair B., “Artificial Intelligence (SIE)”, Mc Graw Hill- 2008. (Units-I,II,IV & V)
2. Dan W. Patterson, “Introduction to AI and ES”, Pearson Education, 2007. (Unit-III). Peter Jackson, “Introduction to Expert Systems”, 3rd Edition, Pearson Education, 2007.
3. Stuart Russel and Peter Norvig “AI – A Modern Approach”, 2nd Edition, Pearson Education 2007.
4. Deepak Khemani “Artificial Intelligence”, Tata Mc Graw Hill Education 2013.

#### **Web Sites**

1. <http://nptel.ac.in>

**18ITU501B****SOFTWARE TESTING****Semester – V  
4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100  
End Semester Exam : 3 Hours****Course Objectives (CO)**

- To learn the basic software debugging methods.
- To understand the White box and Black Box testing methods and techniques
- To design test plans.
- To discuss various software testing issues and solutions in software unit test, integration and system testing
- To learn the different testing tools (familiar with open source tools)
- To understand Quality Assurance models.

**Course Outcomes (COs)**

After completion of this course, the students will be able to

1. Investigate the reason for bugs and analyze the principles in software testing to prevent and remove bugs.
2. Implement various test processes for quality improvement
3. Design test planning.
4. Manage the test process
5. Apply the software testing techniques in commercial environment
6. Use practical knowledge of a variety of ways to test software and an understanding of some of the tradeoffs between testing techniques.

**Unit I Testing Fundamentals**

Examining the Specification: Getting started – Performing a high-level review of the specification – Low-level specification test techniques. Testing the software with blinders on: Dynamic Black-Box Testing- Test-to-Pass and Test-to-Fail- Equivalence Partitioning- Data testing – State testing – Other Black-box test techniques.

**Unit II Examining the code**

Static White-Box testing- Formal reviews – Coding Standards and Guidelines- Generic Code Review Checklist. Testing the software with X-Ray glasses: Dynamic White-Box testing- Dynamic White-Box testing versus Debugging-Testing the Pieces- Data Coverage- Code Coverage.

**Flowgraphs and Path Testing**

Path-testing Basics – Predicates, Path Predicates and Achievable Paths-Path sensitizing-Path Instrumentation-Implementation and Application of Path Testing

**Unit III Transaction-Flow Testing and Data-Flow Testing**

Transaction Flows-Transaction Flow Testing Techniques. Data-Flow Testing Basics-Data-Flow Testing Strategies-Application, Tools, Effectiveness

**Unit IV Domain Testing**

Domains and Paths-Domain Testing-Domains and Interface Testing-Domains and Testability

**Unit V Logic-Based Testing and State Graphs**

Motivational Overview-Decision Tables-Path Expressions Again-KV Charts-Specifications  
State Graphs-Good State Graphs and Bad-State Testing

**Suggested Readings**

1. Boris Beizer (2009), Software Testing Techniques (2<sup>nd</sup> ed.). New Delhi Dreamtech Press
2. Ron Patton (2002) Software Testing (2<sup>nd</sup> ed.). New Delhi: Pearson Education
3. Dorothy Graham, Erik Van Veenendaal, Isabel Evans, Rex Black (2007). Foundations of Software Testing, ISTQB Certification.
4. Brian Hambling, Peter Morgan, Angelina Samaroo, Geoff Thompson (2010). Software Testing , (2<sup>nd</sup> ed.). An ISEB Foundation, BCS
5. Renu Rajani, Pradeep Oak (2004). Software Testing- Effective Methods, Tools and Techniques, Tata McGraw Hill, New Delhi

**Web Sites**

1. [www.testinggeek.com](http://www.testinggeek.com)
2. [www.softwaretestinghelp.com](http://www.softwaretestinghelp.com)
3. [www.softwaretestinginstitute.com](http://www.softwaretestinginstitute.com)



**18ITU502A****. NET PROGRAMMING****Semester – V  
4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours**Course Objectives (CO)**

- To design, formulate, and construct applications with VB.NET
- To integrate variables and constants into calculations applying VB.NET
- To determine logical alternatives with VB.NET decision structures
- To implement lists and loops with VB.NET controls and iteration
- To separate operations into appropriate VB.NET procedures and functions
- To assemble multiple forms, modules, and menus into working VB.NET solutions

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Grasp the fundamentals of a programming language and know the basic differences between programming languages
2. Choose the architecture based on the problem to be solved.
3. Differentiate between the types of applications supported by .Net
4. Build, compile and execute a VB .Net program
5. Apply techniques to develop error-free software
6. To build integrated VB.NET solutions using files and structures with printing capabilities. Translate general requirements into data-related solutions using database concepts

**Unit I**

**Introduction to .NET:** .NET framework features & architecture, CLR, common Type system, MSIL, Assemblies and class libraries. Introduction to visual studio, Project basics, types of project in .Net, IDE of VB .Net – Menu bar, Tool bar, Solution Explorer, Toolbox, Properties Window, Form Designer, Output Window, Object browser. The environment: Editor tab, format tab, general tab, docking tab. Visual development & event driven programming – Methods and events.

**Unit II**

**The VB .Net Language:** The VB .Net Language – Variables- declaring variables, Data type of variables, forcing variables declarations, Course Objectives& lifetime of a variable, constants, arrays, types of arrays, control array, Structure programming – Modularity – Information hiding – abstraction – events – subroutines and functions – message box – input box. Control flow statement: conditional statement, loop statement.

**Unit III**

**Working with WPF:** Introduction: Understanding Windows Graphics – WPF: A Higher Level API – The architecture of WPF. XAML: Basics, properties and events in XAML – loading and compiling – Layout. Classic controls: The Control class – content controls – text controls – list controls – Range based controls.

**Unit IV**

**Objects and Collections:** Understanding objects, properties, methods. Understanding collections. Files: Introduction – classification of files – processing files – handling files and folder using class – directory class – file class.

### Unit V

**Database programming with ADO .Net:** overview of ADO, from ADO to ADO .Net, accessing data using server explorer. Creating connection, command, data adapter and data set with OLEDB and SQLDB. Display data on data bound controls, display data on a data grid. Generate reports using Crystal Report Viewer.

### Suggested Readings

1. Shrishchavan (2007). Visual Basic .Net (1st ed.). New Delhi: Pearson education.
2. Bryan Newsome (2012). Beginning Visual Basic. John Wiley & Sons, Inc.
3. Matthew MacDonald Pro (2008). Windows Presentation Foundation with .Net 3.5 Apress
4. Duncan Mackenzie and Kent Sharkey (2006). Sams Teach Yourself Visual Basic .Net (1st ed.). New Delhi: Techmedia.
5. Ian Griffiths, Chris Shells (2005). Programming Windows Presentation Foundation (1st ed.). O'Reilly Publishers
6. Jeffrey R.Shapiro (2002). The Complete Reference Visual Basic .Net. New Delhi: Tata McGraw Hill Ed.

### Websites

1. [www.startvbdotnet.com](http://www.startvbdotnet.com)
2. [www.functionx.com](http://www.functionx.com)
3. [www.dotnetspider.com](http://www.dotnetspider.com)
4. [www.developerfusion.com](http://www.developerfusion.com)
5. <http://www.wdftutorial.net/HelloWPF.html>

**Semester – V****18ITU502B****NETWORK PROGRAMMING****4H – 4C**

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To understand the 3-tier software architecture (presentation/client tier, application tier, data tier).
- To write web applications using a combination of client-side (JavaScript, HTML, XML, WML) and server-side technologies (JSP, JSF, SERVLETS).
- To write network applications using state-of-the-art RPC technologies including: RMI, CORBA, EJB, and Web Services (SOAP and UDDI).
- To understand e-mail programming (JavaMail, SMTP, POP, IMAP).
- To design and implement network applications through semester-long projects.
- To understand network routing (static and dynamic) and understand the process of implementing simple routed inter-networks.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Analyze the various transmission media, their comparative study, fiber optics and wireless media
2. Categorize the topologies of networks (LAN and WAN), Layered architecture (OSI and TCP/IP) and protocol suites.
3. TCP, UDP, SCTP protocols Ethernet and LAN administration.
4. Details of IP operations in the INTERNET and associated routing principles
5. Understand the key protocols which support the Internet
6. reate applications using techniques such as multiplexing, forking, multithreading

**Unit I**

Transport Layer Protocols: TCP, UDP, SCTP protocol.

**Unit II**

Socket Programming: Socket Introduction; TCP Sockets; TCP Client/Server Example ; signal handling

**Unit III**

I/O multiplexing using sockets; Socket Options; UDP Sockets; UDP client server example; Address lookup using sockets.

**Unit IV**

Network Applications: Remote logging; Email; WWW and HTTP.

**Unit V**

LAN administration: Linux and TCP/IP networking: Network Management and Debugging.

**Suggested Readings**

1. Richard Stevens, W., Bill Fenner., & Andrew, M. Rudoff. (2003). Unix Network Programming, The sockets Networking API, Vol. 1(3rd ed.). New Delhi: PHI.
2. Forouzan, B. A. (2003). Data Communications and Networking(4th ed.). New Delhi: THM Publishing Company Ltd.,
3. Nemeth Synder., & Hein. (2010). Linux Administration Handbook (2nd ed.), New Delhi: Pearson Education.
4. Steven, R. (1990). Unix Network Programming (2nd ed.). New Delhi: PHI.

**Semester – V****18ITU503A****DATA MINING****4H – 4C**

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To be familiar with mathematical foundations of data mining tools.
- To understand and implement classical models and algorithms in data warehouses and data mining
- To characterize the kinds of patterns that can be discovered by association rule mining, classification and clustering.
- To master data mining techniques in various applications like social, scientific and environmental context.
- To develop skill in selecting the appropriate data mining algorithm for solving practical problems
- To develop research interest towards advances in data mining

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Introduce students to the basic concepts and techniques of Data Mining.
2. Develop skills of using recent data mining software for solving practical problems.
3. Gain experience of doing independent study and research.
4. Possess some knowledge of the concepts and terminology associated with database systems, statistics, and machine learning
5. Identify appropriate data mining algorithms to solve real world problems
6. Benefit the user experiences towards research and innovation. integration

**Unit I**

**Overview:** Predictive and descriptive data mining techniques

**Unit II**

Supervised and unsupervised learning techniques

**Unit III**

Process of knowledge discovery in databases, pre-processing methods

**Unit IV**

**Data Mining Techniques:** Association Rule Mining, classification and regression techniques, clustering

**Unit V**

Scalability and data management issues in data mining algorithms, measures of interestingness.

**Suggested Readings**

1. Pang-Ning Tan., Michael Steinbach., & Vipin Kumar. (2005). Introduction to Data Mining. New Delhi: Pearson Education.
2. Richard Roiger., & Michael Geatz. (2003). Data Mining: A Tutorial Based Primer. New Delhi: Pearson Education.
3. Gupta, G.K. (2006). Introduction to Data Mining with Case Studies. New Delhi: PHI.
4. Soman, K. P., Diwakar Shyam., & Ajay, V. (2006). Insight Into Data Mining: Theory And Practice. New Delhi: PHI.

**Web Sites**

1. Thedacs.Com
2. Dwreview.Com
3. Pcai.Com
4. Eruditionhome.Com

**18ITU503B****MACHINE LEARNING****Semester – V**  
**4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours**Course Objectives (CO)**

- To develop an appreciation for what is involved in learning from data.
- To understand a wide variety of learning algorithms.
- To understand how to perform evaluation of learning algorithms and model selection.
- To introduce students to the basic concepts and techniques of Machine Learning.
- To become familiar with regression methods, classification methods, clustering methods.
- To become familiar with Dimensionality reduction Techniques.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Have a broad understanding of machine learning algorithms and their use in data-driven knowledge discovery and program synthesis.
2. Design and implement several machine learning algorithms in Java.
3. Identify, formulate and solve machine learning problems that arise in practical applications.
4. Have knowledge of the strengths and weaknesses of different machine learning algorithms (relative to the characteristics of the application domain) and be able to adapt or combine some of the key elements of existing machine learning algorithms to design new algorithms as needed.
5. Identify machine learning techniques suitable for a given problem
6. Solve the problems using various machine learning techniques

**Unit I**

Introduction: Concept of Machine Learning, Applications of Machine Learning, Key elements of Machine Learning, Supervised vs. Unsupervised Learning, Statistical Learning: Bayesian Method, The Naive Bayes Classifier

**Unit II**

Softwares for Machine Learning and Linear Algebra Overview : Plotting of Data, Vectorization, Matrices and Vectors: Addition, Multiplication, Transpose and Inverse using available tool such as MATLAB.

### **Unit III**

Linear Regression: Prediction using Linear Regression, Gradient Descent, Linear Regression with one variable, Linear Regression with multiple variables, Polynomial Regression, Feature Scaling/Selection.

Logistic Regression: Classification using Logistic Regression, Logistic Regression vs. Linear Regression, Logistic Regression with one variable and with multiple variables.

### **Unit IV**

Regularization: Regularization and its utility: The problem of Overfitting, Application of Regularization in Linear and Logistic Regression, Regularization and Bias/Variance.

### **Unit V**

Neural Networks: Introduction, Model Representation, Gradient Descent vs. Perceptron Training, Stochastic Gradient Descent, Multilayer Perceptrons, Multiclass Representation, Backpropagation Algorithm.

### **Suggested Readings**

1. Santanu Chattopadhyaya. (2011). Systems Programming. New Delhi: PHI.
2. Alfred, V. Aho., Monica, S. Lam., Ravi Sethi., & Jeffrey, D. Ullman. (2006). Compilers: Principles, Techniques, and Tools (2nd ed.). New Delhi: Prentice Hall.
3. Dhamdhare, D. M. (2011). Systems Programming. New Delhi: Tata McGraw Hill.
4. Leland Beck., & Manjula, D. (2008). System Software: An Introduction to System Programming (3<sup>rd</sup> ed.). New Delhi: Pearson Education.
5. Grune, D., Van Reeuwijk, K., Bal, H. E., Jacobs, C. J. H., & Langendoen, K. (2012). Modern Compiler Design (2nd ed.). Springer.



---

<b>18ITU504A</b>	<b>DIGITAL IMAGE PROCESSING</b>	<b>Semester – V</b> <b>3H – 3C</b>
------------------	---------------------------------	---------------------------------------

---

**Instruction Hours / week: L: 0 T: 0 P: 3    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To study the image fundamentals and mathematical transforms necessary for image processing.
- To study the image enhancement techniques
- To study image restoration procedures.
- To study the image compression procedures.
- To introduce to the students the basics of digital image processing.
- To learn the basic image transforms, segmentation algorithms and problems of object measurements.

**Course Outcomes (COs)**

Upon completion of this course, the students will be able to

1. Review the fundamental concepts of a digital image processing system.
2. Analyze images in the frequency domain using various transforms.
3. Evaluate the techniques for image enhancement and image restoration.
4. Categorize various compression techniques.
5. Interpret Image compression standards.
6. Interpret image segmentation and representation techniques.

**Unit I**

Introduction: Light, Brightness adaption and discrimination, Pixels, coordinate conventions, Imaging Geometry, Perspective Projection, Spatial Domain Filtering, sampling and quantization. Spatial Domain Filtering: Intensity transformations, contrast stretching, histogram equalization, Correlation and convolution, Smoothing filters, sharpening filters, gradient and Laplacian.

**Unit II**

Hotelling Transform, Fourier Transforms and properties, FFT (Decimation in Frequency and Decimation in Time Techniques), Convolution, Correlation, 2-D sampling, Discrete Cosine Transform, Frequency domain filtering.

**Unit III**

Image Restoration, Basic Framework, Interactive Restoration, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Adaptive filters, Linear, Position invariant degradations, Image Compression-Encoder-Decoder model, Types of redundancies, Lossy and Lossless compression, Entropy of an information source, Shannon's 1st Theorem, Huffman Coding, Arithmetic Coding, Golomb Coding, LZW coding, Transform Coding, Run length coding.

**Unit IV**

FAX compression (CCITT Group-3 and Group-4), Symbol-based coding, Bit-allocation, Zonal Coding, Threshold Coding, JPEG, Lossless predictive coding, Lossy predictive coding, Motion Compensation

Wavelet based Image Compression: Wavelet series expansion, Discrete Wavelet Transform (DWT), Continuous Wavelet Transform, Fast Wavelet Transform, 2-D wavelet Transform, JPEG-2000 encoding, Digital Image Watermarking

**Unit V**

Morphological Image Processing: Basics, SE, Erosion, Dilation, Opening, Closing, Boundary Detection, skeletons, pruning. Image Segmentation: Boundary detection based techniques, Point, line detection, Edge detection, Edge linking, local processing, regional processing, Hough transform, Thresholding, Iterative thresholding.

**Suggested Readings**

1. Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing(3rd ed.). New Delhi: Pearson Education.
2. Jain, A. K. (1989). Fundamentals of Digital image Processing. New Delhi: Prentice Hall of India.
3. Castleman, K. R. (1996). Digital Image Processing. New Delhi: Pearson Education.
4. Schalkoff. (1989). Digital Image Processing and Computer Vision. New York: John Wiley and Sons.
5. Rafael, C. Gonzalez., Richard, E. Woods., & Steven Eddins. (2004). Digital Image Processing using MATLAB. New Delhi: Pearson Education.

**18ITU504B****MULTIMEDIA AND ITS APPLICATIONS****Semester – V  
3H – 3C****Instruction Hours / week: L: 0 T: 0 P: 3    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours****Course Objectives (CO)**

- To understand the multimedia communications systems, application and basic principles,
- To analyze of the multimedia streaming,
- To perform and establish multimedia communication terminals,
- To present multimedia communications
- Explore a brief history of multimedia in education
- Analyze instructional and informational media (print materials, audio/visual materials and/or web-based materials, games/simulations, etc.)

**Course Outcomes (COs)**

Upon successful completion the student will be able to:

1. Define multimedia to potential clients.
2. Identify and describe the function of the general skill sets in the multimedia industry.
3. Identify the basic components of a multimedia project.
4. Identify the basic hardware and software requirements for multimedia development and playback.
5. Describe the process of digitizing (quantization) of different analog signals (text, graphics, sound and video).
6. Use appropriate tools for the design, development and creation of digital media artefacts.

**Unit I**

**Multimedia – An overview:** Introduction – Multimedia presentation and production – Characteristics of Multimedia presentation – Hardware and Software requirements – Uses of Multimedia. Text: Types of text – Font- Text file formats. Image: Image data representation – Image file formats – image processing software. Graphics: Advantages of graphics – Uses – Component of a graphics system.

**Unit II**

**Audio:** Sound waves – types and properties of sound – components of audio system – Digital audio – Musical Instrument Digital Interface (MIDI) – Audio file formats – Audio processing software. Video: Motion video – Television systems – Video file formats – video processing software. Animation: Uses of animation – computer based animation -Animation file formats – Animation software.

**Unit III**

**Introducing photoshop elements:** About elements – welcome screen – create mode – menu bar – toolbox – options bar – panels. Organizing images: Obtaining images -tagging images – searching for images – opening and saving images. Selecting areas – Layers – Text and drawing tools.

**Unit IV**

**Understanding flash:** Understanding flash basic elements – creating a simple animation. Learning Flash toolbox: Learning the toolbox – using tools. Learning flash panels: Understanding the panels. Using timeline and layers: Understanding how timeline works – Understanding layers. Drawing objects: Drawing lines and fills – using colors – Rotating, skewing and scaling – grouping objects. Creating animation – How animation works – creating motion tweens – creating shape tweens. Understanding masks – creating masks.

## Unit V

**Creating symbols and using the library:** Learning about symbols – creating symbols – using libraries. Learning Basic Actionscript concepts: Actionscript basics – data type basics. Learning basic actionscript programming: Applying Actionscript – Using Actionscript to control actions – Using Actionscript to control properties – Understanding Actions and Event Handlers.

## Suggested Readings

1. Ranjan Parekh (2013). Principles of Multimedia (2nd ed.). TataMcGraw Hill.
2. Nick Vandome (2011). Photoshop Elements 9. TataMcGraw Hill.
3. Brian Underdahl (2002). Macromedia Flash MX – A Beginners Guide. Dreamtech Press.
4. Tay Vaughan (2002). Fundamentals of Multimedia (5th ed.). TataMcGraw Hill.
5. Bill Sanders (2001). Flash 5 Actionscript (1st ed.). New Delhi DreamTech Press.

## Websites

1. [en.wikipedia.org/wiki/Multimedia](http://en.wikipedia.org/wiki/Multimedia)
2. [www.arena-multimedia.com/](http://www.arena-multimedia.com/)
3. [www.nextwavemultimedia.com/](http://www.nextwavemultimedia.com/)

## Semester – V

18ITU511A

ARTIFICIAL INTELLIGENCE - PRACTICAL

4H – 2C

**Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours

**Course Objectives (CO)**

- To understand the various characteristics of Intelligent agents
- To learn about the different search strategies in AI
- To learn to represent knowledge in solving AI problems
- To understand the different ways of designing software agents
- To know about the various applications of AI.
- Understand the various searching techniques, constraint satisfaction problem and example problems- game playing techniques.

**Course Outcomes (COs)**

At the end of the course, the student should be able to:

1. Identify problems that are amenable to solution by AI methods.
2. Identify appropriate AI methods to solve a given problem.
3. Formalize a given problem in the language/framework of different AI methods.
4. Implement basic AI algorithms.
5. Design and carry out an empirical evaluation of different algorithms on a problem formalization, and state the conclusions that the evaluation supports.
6. Ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems

**List of Programs**

Write the following programs using PROLOG

1. Program to add two numbers.
2. Program to categorize animal characteristics.
3. Program to read address of a person using compound variable.
4. Program of fun to show concept of cut operator .
5. Program to count number of elements in a list .
6. Program to reverse the list.
7. Program to append an integer into the list .
8. Program to replace an integer from the list .
9. Program to delete an integer from the list .
10. Program to show concept of list.
11. Program to demonstrate family relationship.
12. Program to show how integer variable is used in prolog program .
13. Write a program to solve 8 queens problem
14. Solve any problem using depth first search.
15. Solve any problem using best first search.
16. Solve traveling salesman problem.

**Suggested Readings**

1. Kevin Night and Elaine Rich, Nair B., “Artificial Intelligence (SIE)”, Mc Graw Hill- 2008. (Units-I,II,IV & V)
2. Dan W. Patterson, “Introduction to AI and ES”, Pearson Education, 2007. (Unit-III).  
Peter Jackson, “Introduction to Expert Systems”, 3rd Edition, Pearson Education, 2007.
3. Stuart Russel and Peter Norvig “AI – A Modern Approach”, 2nd Edition, Pearson Education 2007.
4. Deepak Khemani “Artificial Intelligence”, Tata Mc Graw Hill Education 2013.

**Web Sites**

2. <http://nptel.ac.in>

**Semester – V****18ITU511B****SOFTWARE TESTING - PRACTICAL****4H – 2C**

**Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To understand the nature of software development and software life cycle process models, agile software development, SCRUM and other agile practices.
- To explain methods of capturing, specifying, visualizing and analyzing software requirements.
- To understand concepts and principles of software design and user-centric approach and principles of effective user interfaces. 4. To know basics of testing and understanding concept of software quality assurance and software configuration management process.
- To understand the need of project management and project management life cycle.
- To understand project scheduling concept and risk management associated to various type of projects.
- Implement a given software design using sound development practices.

**Course Outcomes (COs)**

1. Apply their knowledge of mathematics, sciences, and computer science to the modeling, analysis, and measurement of software artifacts.
2. Work effectively as leader/member of a development team to deliver quality software artifacts.
3. Analyze, specify and document software requirements for a software system.
4. Verify, validate, assess and assure the quality of software artifacts.
5. Design, select and apply the most appropriate software engineering process for a given project, plan for a software project, identify its Course scope and risks, and estimate its cost and time.
6. Express and understand the importance of negotiation, effective work habits, leadership, and good communication with stakeholders, in written and oral forms, in a typical software development environment.

**List of Programs**

1. Write a program that take three inputs (a,b and c) that represents the sides of a triangle and the output is one of the below four.
  - a) Not a triangle
  - b) Scalene triangle
  - c) Isosceles triangle
  - d) Equilateral triangle
- a) Generate test cases using boundary value analysis, equivalence class partitioning and decision table testing
- b) Generate test cases using basis path testing
- c) Run code coverage tool

2. Write a program that determines the nature of roots of a quadratic equation. Output should be one of the following
  - Not a quadratic equation
  - Complex roots
  - Real roots
  - Single roots
  - a) Generate test cases using boundary value analysis, equivalence class partitioning and decision table testing
  - b) Generate test cases using basis path testing
  - c) Run code coverage tool
3. Write a program that checks whether the number is even or odd. Run code coverage tools and find the amount of code being covered.
4. Write a program that dynamically allocates memory to 10 integers using malloc() or calloc() and do not free memory leading to memory leaks. Verify the same using Valgrind. Now, free memory using free() at the end of the program to avoid memory leaks. Verify the same using Valgrind.
5. Using Selenium IDE, write a test suite containing minimum 4 test cases.
6. Conduct a test suite for any two websites.
7. Write and test a program to login a specific webpage

### Suggested Readings

1. Boris Beizer (2009), Software Testing Techniques (2<sup>nd</sup> ed.). New Delhi Dreamtech Press
2. Ron Patton (2002) Software Testing (2<sup>nd</sup> ed.). New Delhi: Pearson Education
3. Dorothy Graham, Erik Van Veenendaal, Isabel Evans, Rex Black (2007). Foundations of Software Testing, ISTQB Certification.
4. Brian Hambling, Peter Morgan, Angelina Samaroo, Geoff Thompson (2010). Software Testing , (2<sup>nd</sup> ed.). An ISEB Foundation, BCS
5. Renu Rajani, Pradeep Oak (2004). Software Testing- Effective Methods, Tools and Techniques, Tata McGraw Hill, New Delhi

### Web Sites

1. [www.testinggeek.com](http://www.testinggeek.com)
2. [www.softwaretestinghelp.com](http://www.softwaretestinghelp.com)
3. [www.softwaretestinginstitute.com](http://www.softwaretestinginstitute.com)



**18ITU512A****.NET PROGRAMMING - PRACTICAL****Semester – V  
4H – 2C****Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours**Course Objectives (CO)**

- To design, formulate, and construct applications with VB.NET
- To integrate variables and constants into calculations applying VB.NET
- To determine logical alternatives with VB.NET decision structures
- To implement lists and loops with VB.NET controls and iteration
- To separate operations into appropriate VB.NET procedures and functions
- To assemble multiple forms, modules, and menus into working VB.NET solutions

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Grasp the fundamentals of a programming language and know the basic differences between programming languages
2. Choose the architecture based on the problem to be solved.
3. Differentiate between the types of applications supported by .Net
4. Build, compile and execute a VB .Net program
5. Apply techniques to develop error-free software
6. To build integrated VB.NET solutions using files and structures with printing capabilities.  
Translate general requirements into data-related solutions using database concepts

**List of Programs**

1. Write a VB .Net program to calculate simple interest and compound interest.
2. Write a VB .Net program to implement Calculator.
3. Write a VB.Net program to implement Notepad
4. Write a VB.Net program to draw several shaps and fill with color.
5. Write a VB .Net program to perform the following in list box
  - a) Add an item
  - b) Delete an item
  - c) List count
  - d) Clear the list
6. Write a program to calculate the total marks of the student and print the grades.
7. Write a VB .Net program to implement employee payroll system
8. Write a VB .Net program to create and manipulate a file.
9. Write a program to implement a web browser
10. Write a program to maintain the details of doctors in a hospital with their specializations

11. Write a program to animate the picture using timer control.
12. Write a program to move the object from one location to another. Change the color and size of object at different time interval.
13. Write a program to place 10 pictures in the listbox. Using timer control the take the picture from listbox and change the form background after specific time interval.
14. Write a program to implement speaking program. Get the text input from the user and convert into voice.
15. Write a program to implement chatting

### **Suggested Readings**

1. Shrishchavan (2007). Visual Basic .Net (1st ed.). New Delhi: Pearson education.
2. Bryan Newsome (2012). Beginning Visual Basic. John Wiley & Sons, Inc.
3. Matthew MacDonald Pro (2008). Windows Presentation Foundation with .Net 3.5 Apress
4. Duncan Mackenzie and Kent Sharkey (2006). Sams Teach Yourself Visual Basic .Net (1st ed.). New Delhi: Techmedia.
5. Ian Griffiths, Chris Shells (2005). Programming Windows Presentation Foundation (1st ed.). O'Reilly Publishers
6. Jeffrey R.Shapiro (2002). The Complete Reference Visual Basic .Net. New Delhi: Tata McGraw Hill Ed.

### **Websites**

1. [www.startvb.net](http://www.startvb.net)
2. [www.functionx.com](http://www.functionx.com)
3. [www.dotnetspider.com](http://www.dotnetspider.com)
4. [www.developerfusion.com](http://www.developerfusion.com)
5. [http://www.wdftutorial.net/HelloWPF.html](http://www.wdftutorial.net>HelloWPF.html)

---

<b>18ITU512B</b>	<b>NETWORK PROGRAMMING - PRACTICAL</b>	<b>Semester – V</b> <b>4H – 2C</b>
------------------	--	---------------------------------------

---

**Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours

**Course Objectives (CO)**

- To understand the 3-tier software architecture (presentation/client tier, application tier, data tier).
- To write web applications using a combination of client-side (JavaScript, HTML, XML, WML) and server-side technologies (JSP, JSF, SERVLETS).
- To write network applications using state-of-the-art RPC technologies including: RMI, CORBA, EJB, and Web Services (SOAP and UDDI).
- To understand e-mail programming (JavaMail, SMTP, POP, IMAP).
- To design and implement network applications through semester-long projects.
- To understand network routing (static and dynamic) and understand the process of implementing simple routed inter-networks.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Analyze the various transmission media, their comparative study, fiber optics and wireless media
2. Categorize the topologies of networks (LAN and WAN), Layered architecture (OSI and TCP/IP) and protocol suites.
3. TCP, UDP, SCTP protocols Ethernet and LAN administration.
4. Details of IP operations in the INTERNET and associated routing principles
5. Understand the key protocols which support the Internet
6. Create applications using techniques such as multiplexing, forking, multithreading

**List of Programs**

1. Understanding and using of commands like ifconfig, netstat, ping, arp, telnet, ftp, finger, traceroute, whois
2. Socket Programming: Implementation of Connection-Oriented Service using standard ports.
3. Implementation of Connection-Less Service using standard ports
4. Implementation of Connection-Oriented Iterative Echo-Server, date and time, character generation using user-defined ports
5. Implementation of Connectionless Iterative Echo-server, date and time, character generation using user-defined ports.
6. Implementation of Connection-Oriented Concurrent Echo-server, date and time, character generation using user-defined ports
7. Program for connection-oriented Iterative Service in which server reverses the string sent by the client and sends it back

8. Program for connection-oriented Iterative service in which server changes the case of the strings sent by the client and sends back (Case Server).
9. Program for Connection-Oriented Iterative service in which server calculates the Net-salary of an Employee based on the following details sent by the client i)basic-sal ii) hra iii) da iv) pt v) epf ( net-sala=basic+hra+da-pt-epf)
10. Implementation of concurrent chat server that allows current logged in users to communicate one with other.

### **Suggested Readings**

1. Richard Stevens, W., Bill Fenner., & Andrew, M. Rudoff. (2003). Unix Network Programming, The sockets Networking API, Vol. 1(3rd ed.). New Delhi: PHI.
2. Forouzan, B. A. (2003). Data Communications and Networking(4th ed.). New Delhi: THM Publishing Company Ltd.,
3. Nemeth Synder., & Hein. (2010). Linux Administration Handbook (2nd ed.), New Delhi: Pearson Education.
4. Steven, R. (1990). Unix Network Programming (2nd ed.). New Delhi: PHI.

**18ITU513A****DATA MINING - PRACTICAL****Semester – V  
4H – 2C****Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours**Course Objectives (CO)**

- To be familiar with mathematical foundations of data mining tools.
- To understand and implement classical models and algorithms in data warehouses and data mining
- To characterize the kinds of patterns that can be discovered by association rule mining, classification and clustering.
- To master data mining techniques in various applications like social, scientific and environmental context.
- To develop skill in selecting the appropriate data mining algorithm for solving practical problems
- To develop research interest towards advances in data mining

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Introduce students to the basic concepts and techniques of Data Mining.
2. Develop skills of using recent data mining software for solving practical problems.
3. Gain experience of doing independent study and research.
4. Possess some knowledge of the concepts and terminology associated with database systems, statistics, and machine learning
5. Identify appropriate data mining algorithms to solve real world problems
6. Benefit the user experiences towards research and innovation. integration

**List of Programs**

1. Use the following learning schemes, with the default settings to analyze the weather data (in weather.arff). for test options, first choose “Use training set”, then choose “Percentage split” using default 66% percentage split. Report model percent error rate.
2. Use iris dataset preprocess and classify it with j4.8 and Naive Bayes classifier. Examine the tree in the classifier output panel.
3. Using the dataset ReutersCorn – Train and ReutersGrain – Train. Classify articles using binary attributes and word count attributes.
4. Apply any two association rule based algorithm for the supermarket analysis.
5. Using weka experimenter perform comparison analysis of j4.8, oneR and ID3 for vote dataset.
6. Using weka experimenter perform comparison analysis of Naive Bayes with different datasets.
7. Apply ZeroR, OneR and j4.8, to classify the iris data in an experiment using 10 train and test runs, with 66% of the data used for 34% used for testing.
8. Using Weka Knowledge flow set up a flow to load an ARFF file (batch mode) and perform a cross-validation using j4.8 (WEKS’s C4.5 implementation).

9. Draw multiple ROC curves in the same plot window, using j4.8 and RandomForest as classifiers.
10. Use any three clustering algorithm on Vehicle data set and find best among them.

**Suggested Readings**

1. Pang-Ning Tan., Michael Steinbach., & Vipin Kumar. (2005). Introduction to Data Mining. New Delhi: Pearson Education.
2. Richard Roiger., & Michael Geatz. (2003). Data Mining: A Tutorial Based Primer. New Delhi: Pearson Education.
3. Gupta, G.K. (2006). Introduction to Data Mining with Case Studies. New Delhi: PHI.
4. Soman, K. P., Diwakar Shyam., & Ajay, V. (2006). Insight Into Data Mining: Theory And Practice. New Delhi: PHI.

**Web Sites**

1. Thedacs.Com
2. Dwreview.Com
3. Pcai.Com
4. Eruditionhome.Com

**Semester – V****18ITU513B****MACHINE LEARNING - PRACTICAL****4H – 2C**

**Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours

**Course Objectives (CO)**

- To develop an appreciation for what is involved in learning from data.
- To understand a wide variety of learning algorithms.
- To understand how to perform evaluation of learning algorithms and model selection.
- To introduce students to the basic concepts and techniques of Machine Learning.
- To become familiar with regression methods, classification methods, clustering methods.
- To become familiar with Dimensionality reduction Techniques.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to:

1. Have a broad understanding of machine learning algorithms and their use in data-driven knowledge discovery and program synthesis.
2. Design and implement several machine learning algorithms in Java.
3. Identify, formulate and solve machine learning problems that arise in practical applications.
4. Have knowledge of the strengths and weaknesses of different machine learning algorithms (relative to the characteristics of the application domain) and be able to adapt or combine some of the key elements of existing machine learning algorithms to design new algorithms as needed.
5. Identify machine learning techniques suitable for a given problem
6. Solve the problems using various machine learning techniques

**List of Programs**

1. Perform elementary mathematical operations in Octave/MATLAB like addition, multiplication, division and exponentiation.
2. Perform elementary logical operations in Octave/MATLAB (like OR, AND, Checking for Equality, NOT, XOR).
3. Create, initialize and display simple variables and simple strings and use simple formatting for variable.

4. Create/Define single dimension / multi-dimension arrays, and arrays with specific values like array of all ones, all zeros, array with random values within a range, or a diagonal matrix.
5. Use command to compute the size of a matrix, size/length of a particular row/column, load data from a text file, store matrix data to a text file, finding out variables and their features in the current Course Objectives.
6. Perform basic operations on matrices (like addition, subtraction, multiplication) and display specific rows or columns of the matrix.
7. Perform other matrix operations like converting matrix data to absolute values, taking the negative of matrix values, adding/removing rows/columns from a matrix, finding the maximum or minimum values in a matrix or in a row/column, and finding the sum of some/all elements in a matrix.
8. Create various type of plots/charts like histograms, plot based on sine/cosine function based on data from a matrix. Further label different axes in a plot and data in a plot.
9. Generate different subplots from a given plot and color plot data.
10. Use conditional statements and different type of loops based on simple example/s.
11. Perform vectorized implementation of simple matrix operation like finding the transpose of a matrix, adding, subtracting or multiplying two matrices.
12. Implement Linear Regression problem. For example, based on a dataset comprising of existing set of prices and area/size of the houses, predict the estimated price of a given house.
13. Based on multiple features/variables perform Linear Regression. For example, based on a number of additional features like number of bedrooms, servant room, number of balconies, number of houses of years a house has been built – predict the price of a house.
14. Implement a classification/ logistic regression problem. For example based on different features of students data, classify, whether a student is suitable for a particular activity. Based on the available dataset, a student can also implement another classification problem like checking whether an email is spam or not.
15. Use some function for regularization of dataset based on problem 14.
16. Use some function for neural networks, like Stochastic Gradient Descent or back propagation - algorithm to predict the value of a variable based on the dataset of problem



**Suggested Readings**

1. Santanu Chattopadhyaya. (2011). Systems Programming. New Delhi: PHI.
2. Alfred, V. Aho., Monica, S. Lam., Ravi Sethi., & Jeffrey, D. Ullman. (2006). Compilers: Principles, Techniques, and Tools (2nd ed.). New Delhi: Prentice Hall.
3. Dhamdhere, D. M. (2011). Systems Programming. New Delhi: Tata McGraw Hill.
4. Leland Beck., & Manjula, D. (2008). System Software: An Introduction to System Programming (3<sup>rd</sup> ed.). New Delhi: Pearson Education.
5. Grune, D., Van Reeuwijk, K., Bal, H. E., Jacobs, C. J. H., & Langendoen, K.(2012). Modern Compiler Design (2nd ed.). Springer.

---

<b>18ITU514A</b>	<b>DIGITAL IMAGE PROCESSING - PRACTICAL</b>	<b>Semester – V</b>
		<b>3H – 1C</b>

---

**Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To study the image fundamentals and mathematical transforms necessary for image processing.
- To study the image enhancement techniques
- To study image restoration procedures.
- To study the image compression procedures.
- To introduce to the students the basics of digital image processing.
- To learn the basic image transforms, segmentation algorithms and problems of object measurements.

**Course Outcomes (COs)**

Upon completion of this course, the students will be able to

1. Review the fundamental concepts of a digital image processing system.
2. Analyze images in the frequency domain using various transforms.
3. Evaluate the techniques for image enhancement and image restoration.
4. Categorize various compression techniques.
5. Interpret Image compression standards.
6. Interpret image segmentation and representation techniques.

**List of Programs**

1. Write program to read and display digital image using MATLAB or SCILAB
  - a. Become familiar with SCILAB/MATLAB Basic commands
  - b. Read and display image in SCILAB/MATLAB
  - c. Resize given image
  - d. Convert given color image into gray-scale image
  - e. Convert given color/gray-scale image into black & white image
  - f. Draw image profile
  - g. Separate color image in three R G & B planes
  - h. Create color image using R, G and B three separate planes
  - i. Flow control and LOOP in SCILAB
  - j. Write given 2-D data in image file
2. To write and execute image processing programs using point processing method
  - a. Obtain Negative image                      b. Obtain Flip image
  - b. Thresholding                                      d. Contrast stretching
3. To write and execute programs for image arithmetic operations
  - a. Addition of two images
  - b. Subtract one image from other image

- c. Calculate mean value of image
  - d. Different Brightness by changing mean value
4. To write and execute programs for image logical operations
- a. AND operation between two images
  - b. OR operation between two images
  - c. Calculate intersection of two images
  - d. Water Marking using EX-OR operation
  - e. NOT operation (Negative image)
5. To write a program for histogram calculation and equalization using
- a. Standard MATLAB function
  - b. Program without using standard MATLAB functions
  - c. C Program
6. To write and execute program for geometric transformation of image
- a. Translation
  - b. Scaling
  - c. Rotation
  - d. Shrinking
  - e. Zooming
7. To understand various image noise models and to write programs for
- a. image restoration
  - b. Remove Salt and Pepper Noise
  - c. Minimize Gaussian noise
  - d. Median filter and Weiner filter
8. Write and execute programs to remove noise using spatial filters
- a. Understand 1-D and 2-D convolution process
  - b. Use 3x3 Mask for low pass filter and high pass filter
9. Write and execute programs for image frequency domain filtering
- a. Apply FFT on given image
  - b. Perform low pass and high pass filtering in frequency domain
  - c. Apply IFFT to reconstruct image
10. Write a program in C and MATLAB/SCILAB for edge detection using different edge detection mask
11. Write and execute program for image morphological operations erosion and dilation.
12. To write and execute program for wavelet transform on given image and perform inverse wavelet transform to reconstruct image.

**Suggested Readings**

1. Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing(3rd ed.). New Delhi: Pearson Education.
2. Jain, A. K. (1989). Fundamentals of Digital image Processing. New Delhi: Prentice Hall of India.
3. Castleman, K. R. (1996). Digital Image Processing. New Delhi: Pearson Education.
4. Schalkoff. (1989). Digital Image Processing and Computer Vision. New York: John Wiley and Sons.
5. Rafael, C. Gonzalez., Richard, E. Woods.,& Steven Eddins. (2004). Digital Image Processing using MATLAB. New Delhi: Pearson Education.

---

**Semester – V**

**18ITU514B MULTIMEDIA AND ITS APPLICATIONS – PRACTICAL      3H – 1C**

---

**Instruction Hours / week: L: 0 T: 0 P: 4      Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours

**Course Objectives (CO)**

- To understand the multimedia communications systems, application and basic principles,
- To analyze of the multimedia streaming,
- To perform and establish multimedia communication terminals,
- To present multimedia communications
- Explore a brief history of multimedia in education
- Analyze instructional and informational media (print materials, audio/visual materials and/or web-based materials, games/simulations, etc.)

**Course Outcomes (COs)**

Upon successful completion the student will be able to:

1. Define multimedia to potential clients.
2. Identify and describe the function of the general skill sets in the multimedia industry.
3. Identify the basic components of a multimedia project.
4. Identify the basic hardware and software requirements for multimedia development and playback.
5. Describe the process of digitizing (quantization) of different analog signals (text, graphics, sound and video).
6. Use appropriate tools for the design, development and creation of digital media artefacts.

**List of Programs**

Perform the following practical exercises GIMP/ Synfig.

1. To change from one shape to another shape
2. To perform rainy effect
3. To subtract one shape from another shape
4. To perform dreamy effect
5. To perform fractal effect
6. To perform transparent glass lettering
7. To bounce a ball
8. To perform smoky effect
9. To perform text portrait
10. To perform bokeh effect

**Suggested Readings**

1. Ranjan Parekh (2013). Principles of Multimedia (2nd ed.). TataMcGraw Hill.
2. Nick Vandome (2011). Photoshop Elements 9. TataMcGraw Hill.
3. Brian Underdahl (2002). Macromedia Flash MX – A Beginners Guide. Dreamtech Press.
4. Tay Vaughan (2002). Fundamentals of Multimedia (5th ed.). TataMcGraw Hill.
5. Bill Sanders (2001). Flash 5 Actionscript (1st ed.). New Delhi DreamTech Press.

**Websites**

- 1.[en.wikipedia.org/wiki/Multimedia](http://en.wikipedia.org/wiki/Multimedia)
- 2.[www.arena-multimedia.com/](http://www.arena-multimedia.com/)
- 3.[www.nextwavemultimedia.com/](http://www.nextwavemultimedia.com/)

**Semester – VI****18ITU601A****PHP PROGRAMMING****4H – 4C**

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours

**Course Objectives (CO)**

- To understand how server-side programming works on the web.
- To learn PHP Basic syntax for variable types and calculations.
- To use PHP built-in functions and creating custom functions
- To understand POST and GET in form submission.
- To receive and process form submission data.
- To create a database in phpMyAdmin, to read and process data in a MySQL database

**Course Outcomes (COs)**

Upon completion of this course, the students will be able to

1. Write PHP scripts to handle HTML forms.
2. Write regular expressions including modifiers, operators, and metacharacters.
3. Create PHP programs that use various PHP library functions, and that manipulate files and directories.
4. Analyze and solve various database tasks using the PHP language.
5. Analyze and solve common Web application tasks by writing PHP programs
6. Get hands on experience on various techniques of web development and will be able to design and develop a complete website.

**Unit I****Introduction to PHP:**

PHP introduction, inventions and versions, important tools and software requirements (like Web Server, Database, Editors etc.) -PHP with other technologies, Course Objectives of PHP  
Basic Syntax, PHP variables and constants-Types of data in PHP , Expressions, Course Objectives of a variable (local, global)-PHP Operators: Arithmetic, Assignment, Relational, Logical operators, Bitwise , ternary and MOD operator.-PHP operator Precedence and associativity

**Unit II****Handling HTML form with PHP:**

Capturing Form Data -GET and POST form methods- Dealing with multi value fields - Redirecting a form after submission -**PHP conditional events and Loops:** PHP IF Else conditional statements ( Nested IF and Else) -Switch case, while ,For and Do While Loop -Goto , Break ,Continue and exit

**Unit III****PHP Functions:**

Function, Need of Function , declaration and calling of a function-PHP Function with arguments, Default Arguments in Function-Function argument with call by value, call by reference-Course Objectives of Function Global and Local

**Unit IV**

**String Manipulation and Regular Expression: (3L)** Creating and accessing String , Searching & Replacing String-Formatting, joining and splitting String , String Related Library functions - Use and advantage of regular expression over inbuilt function-Use of preg\_match(), preg\_replace(), preg\_split() functions in regular expression.

**Unit V****Array:**

Anatomy of an Array ,Creating index based and Associative array ,Accessing array  
-Looping with Index based array, with associative array using each() and foreach()  
-Some useful Library function

**Suggested Readings**

1. Steven Holzner. (2007). PHP: The Complete Reference. New Delhi: McGraw Hill Education (India).
2. Timothy Boronczyk., & Martin, E. Psinas. (2008). PHP and MYSQL (Create-Modify-Reuse). New Delhi: Wiley India Private Limited.
3. Robin Nixon. (2014). Learning PHP, MySQL, JavaScript, CSS & HTML5 (3rd ed.). O'reilly.
4. Luke Welling.,& Laura Thompson.(2008). PHP and MySQL Web Development (4th ed.). Addition Paperback, Addison-Wesley Professional.
5. David Sklar., & Adam Trachtenberg. PHP Cookbook: Solutions & Examples for PHP.

**Web Sites**

1. <http://www.w3schools.com/php/>



**18ITU601B****UNIX / LINUX PROGRAMMING****Semester – VI  
4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours**Course Objectives (CO)**

- To understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.
- To understand how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions
- To understand how the operating system abstractions can be implemented
- To understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software
- To understand basic resource management techniques (scheduling or time management, space management) and principles and how they can be implemented.
- These also include issues of performance and fairness, avoiding deadlocks, as well as security and protection.

**Course Outcomes (COs)**

Upon completion of this course, the students will be able to

1. Develop software for Linux/UNIX systems.
2. Learn the C language and get experience programming in C.
3. Learn the important Linux/UNIX library functions and system calls.
4. Understand the inner workings of UNIX-like operating systems.
5. Obtain a foundation for an advanced course in operating systems.
6. Construct various shell scripts for simple applications

**Unit I****Introduction** What are Linux/Unix Operating systems, Difference between linux/unix and other operating systems, Features and Architecture, Various Distributions available in the market, Installation, Booting and shutdown process.**Unit II**

System processes (an overview), External and internal commands, Creation of partitions in OS, Processes and its creation phases – Fork, Exec, wait

**Unit III****User Management and the File System** Types of Users, Creating users, Granting rights  
User management commands, File quota and various file systems available, File System Management and Layout, File permissions, Login process, Managing Disk Quotas, Links (hard links, symbolic links)

**Unit IV**

**Shell introduction and Shell Scripting** What is shell and various type of shell, Various editors present in Linux Different modes of operation in vi editor, What is shell script, Writing and executing the shell script , Shell variable (user defined and system variables)

**Unit V**

**System calls, Using system calls** Pipes and Filters, Decision making in Shell Scripts (If else, switch), Loops in shell.

Functions, Utility programs (cut, paste, join, tr, uniq utilities), Pattern matching utility (grep)

**Suggested Readings**

1. Sumitabha, Das.(2006). Unix Concepts And Applications. New Delhi: Tata McGraw-Hill Education.
2. Michael Jang. (2011). RHCSA/ RHCE Red Hat Linux Certification: Exams (Ex200 & Ex300). Certification Press.
3. Nemeth Synder., & Hein.(2010). Linux Administration Handbook (2nd ed.). Pearson Education.
4. Richard Stevens, W. Bill Fenner., & Andrew, M. Rudoff. (2014). Unix Network Programming, The sockets Networking API, Vol. 1, (3rd ed.).

**Semester – VI****18ITU602A****E-COMMERCE TECHNOLOGIES****4H – 4C**

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To introduce the concepts, vocabulary, and procedures associated with E-Commerce and the Internet.
- To gain an overview of all aspects of E-Commerce.
- To develop the Internet and E-Commerce, options available for doing business on the Internet, features of Web sites and the tools used to build an E-Commerce web site, marketing issues, payment options, security issues, and customer service.
- It is designed to familiarize students with organizational and managerial foundations of systems, the technical foundation for understanding information systems
- Assess e-commerce strategies and applications, including online marketing, e-government, e-learning and global e-commerce

**Course Outcomes (COs)**

Upon successful completion of this course, the student will be able to:

1. Describe an example of system architecture for an e-Business.
2. List the seven major elements of web design.
3. Identify and explain fundamental web site tools including design tools, programming tools, and data processing tools.
4. Identify the major electronic payment issues and options.
5. Discuss security issues and explain procedures used to protect against security threats.
6. Identify and discuss management issues underlying e-Commerce issues including organizational structure, strategic planning, goal setting, corporate social responsibility, international arena, changing market intermediaries, resource allocation and customer service

**Unit I -An Introduction to Electronic commerce**

What is E-Commerce (Introduction And Definition), Main activities E-Commerce, Goals of E-Commerce, Technical Components of E-Commerce, Functions of E-Commerce, Advantages and disadvantages of E-Commerce, Course Objectives of E-Commerce, Electronic Commerce Applications, Electronic Commerce and Electronic Business (C2C) (C2G, G2G, B2G, B2P, B2A, P2P, B2A, C2A, B2B, B2C)

**Unit II -The Internet and WWW**

Evolution of Internet, Domain Names and Internet - Organization (.edu, .com, .mil, .gov, .net etc), Types of Network, Internet Service Provider, World Wide Web, Internet & Extranet, Role of Internet in B2B Application, building own website, cost, time, reach, registering a domain name, web promotion, Target email, Banner, Exchange, Shopping Bots.

**Unit III - Electronic data**

Electronic data exchange introduction, concepts of EDI and Limitation, Application of eDI, Disadvantages of eDI, EDI model, Electronic Payment System: Introduction, Types of Electronic Payment system, Payment types, Value exchange system, credit card system electronic fund transfer, Paperless bill, modern payment cash, Electronic cash.

**Unit IV- Planning for Electronic Commerce**

Planning Electronic commerce initiatives, linking Course Outcomes (COs) to business strategies, measuring cost Course Outcomes (COs), comparing benefits to Costs, strategies for developing electronic commerce web sites.

**Unit V - Internet marketing**

The PROS and CONS of online shopping, the CONS of online shopping, Justify an internet business, Internet marketing techniques, The E-Cycle of Internet marketing, personalization e-commerce.

**Suggested Readings**

1. G.S.V. Murthy (2011). E-Commerce concepts, Models, Strategies. Himalaya Publishing house.
2. Gray. P. Schneider (2011). Electronic commerce International student edition.
3. Henry Cahn, Raymond Lee, Tharam Dillon, Elizabeth Chang. (2011). E-Commerce fundamentals and Applications. Wiley Student Edition.
4. Kamlesh K. Bajaj and Debjani Nag (2005). E-Commerce.
5. David Whitley (2000). E-Commerce-strategies, Technologies and Applications. TMH.

**Websites**

1. [http://www.tutorialspoint.com/e\\_commerce/e\\_commerce\\_tutorial.pdf](http://www.tutorialspoint.com/e_commerce/e_commerce_tutorial.pdf)
2. <http://www.dynamicwebs.com.au/tutorials/e-commerce.htm>
3. <http://www.htmlgoodies.com/beyond/webmaster/projects/electronic-commerce-tutorial.html>

**18ITU602B****CLOUD COMPUTING****Semester – VI  
4H – 4C****Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours**Course Objectives (CO)**

- To learn how to use Cloud Services.
- To implement Virtualization
- To implement Task Scheduling algorithms.
- To apply Map-Reduce concept to applications.
- To build Private Cloud.
- To know the impact of engineering on legal and societal issues involved

**Course Outcomes (COs)**

Upon completion of this course, the students will be able to

1. Analyze the Cloud computing setup with its vulnerabilities and applications using different architectures.
2. Design different workflows according to requirements and apply map reduce programming model.
3. Apply and design suitable Virtualization concept, Cloud Resource Management and design scheduling algorithms.
4. CO4: Create combinatorial auctions for cloud resources and design scheduling algorithms for computing clouds
5. Assess cloud Storage systems and Cloud security, the risks involved, its impact and develop cloud application
6. Broadly educate to know the impact of engineering on legal and societal issues involved in addressing the security issues of cloud computing.

**Unit I**

**Overview of Computing Paradigm:** Recent trends in Computing: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing.**Introduction to Cloud Computing:** Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing.

**Unit II**

**Cloud Computing Architecture:** Comparison with traditional computing architecture (client/server), Services provided at various levels, Service Models- Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS), How Cloud Computing Works, Deployment Models- Public cloud, Private cloud, Hybrid cloud, Community cloud, Case study of NIST architecture.

**Unit III**

**Case Studies:** Case study of Service model using Google App Engine, Microsoft Azure, Amazon EC2, Eucalyptus.

**Unit IV**

**Service Management in Cloud Computing:** Service Level Agreements (SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling.

**Unit V**

**Cloud Security:** Infrastructure Security- Network level security, Host level security, Application level security, Data security and Storage- Data privacy and security Issues, Jurisdictional issues raised by Data location, Authentication in cloud computing.

**Suggested Readings**

1. Barrie Sosinsky. (2010). Cloud Computing Bible. New Delhi: Wiley-India,
2. Rajkumar Buyya., James Broberg., & Andrzej, M. Goscinski Wile. Cloud Computing: Principles and Paradigms.
3. Nikos Antonopoulos., & Lee Gillam. (2012). Cloud Computing: Principles, Systems and Applications. Springer.
4. Ronald, L. Krutz., & Russell Dean Vines. (2010). Cloud Security: A Comprehensive Guide to Secure Cloud Computing. New Delhi: Wiley-India.
5. Gautam Shroff. (2010). Enterprise Cloud Computing Technology Architecture Applications. Adobe Reader ebooks available from eBooks.com.
6. Toby Velte., Anthony Velte., & Robert Elsenpeter.(2010). Cloud Computing, A Practical Approach. McGraw Hills.
7. Dimitris, N. Chorafas. (2010). Cloud Computing Strategies. CRC Press.

**Web Sites**

1. [en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing)
2. [www.ibm.com/cloud-computing/in/en/](http://www.ibm.com/cloud-computing/in/en/)
3. [www.oracle.com/CloudComputing](http://www.oracle.com/CloudComputing)
4. [www.microsoft.com/en-us/cloud/default.aspx](http://www.microsoft.com/en-us/cloud/default.aspx)

**18ITU603A****BIG DATA ANALYTICS****Semester – VI  
3H – 3C****Instruction Hours / week: L: 3 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours**Course Objectives (CO)**

- To provide an overview of an exciting growing field of big data analytics.
- To impart to students the skills required to design scalable systems that can accept, store, and analyze large volumes of unstructured data.
- The objective of this course is to ascertain that the students know the fundamental techniques and tools used to design and analyze large volumes of data.
- To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability.
- To enable students to have skills that will help them to solve complex real-world problems in for decision support.
- To understand, and practice big data analytics and machine learning approaches

**Course Outcomes (COs)**

Upon completion of this course, the students will be able to

1. Explain the motivation for big data systems and identify the main sources of Big Data in the real world.
2. Demonstrate an ability to use frameworks like Hadoop, NOSQL to efficiently store retrieve and process Big Data for Analytics.
3. Implement several Data Intensive tasks using the Map Reduce Paradigm
4. Understand the key issues in big data management and its associated applications in intelligent business and scientific computing.
5. Interpret business models and scientific computing paradigms, and apply software tools for big data analytics.
6. Achieve adequate perspectives of big data analytics in various applications like recommender systems, social media applications etc.

**UNIT-I**

Fundamentals of Big Data - The Evolution of Data Management    Understanding the Waves of Managing Data- Defining Big Data - Big Data Management Architecture- The Big Data Journey -Big Data Types-Defining Structured Data-Defining Unstructured Data-Putting Big Data Together.

**UNIT-II**

Big Data Stack- Basics of Virtualization - The importance of virtualization to big data -Server virtualization - Application virtualization - Network virtualization -Processor and memory virtualization - Data and storage virtualization-Abstraction and Virtualization-Implementing Virtualization to Work with Big Data.

**UNIT-III**

Hadoop - Hadoop Distributed File System - Hadoop MapReduce- The Hadoop foundation and Ecosystem.

#### **UNIT-IV**

Big Data Analytics-Text Analytics and Big Data-Customized Approaches for Analysis of Big Data

#### **UNIT-V**

Integrating Data Sources-Real-Time Data Streams and Complex Event Processing-Operationalizing Big Data.

#### **Suggested readings**

1. Judith Hurwitz, Alan Nugent, Fern Halper, Marcia Kaufman, (2013). Big Data For Dummies, Wiley India, New Delhi.
2. Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan, (2012). Harness the Power of Big Data The IBM Big Data Platform, Tata McGraw Hill Publications, New Delhi.
3. Michael Minelli, Michele Chambers, Ambiga Dhiraj, (2013). Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses, Wiley Publications, New Delhi.
4. Zikopoulos, Paul, Chris Eaton, (2011). Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, Tata McGraw Hill Publications, New Delhi.

#### **Websites**

1. [www.oracle.com/BigData](http://www.oracle.com/BigData)
2. [www.planet-data.eu/sites/default/files/Big\\_Data\\_Tutorial\\_part4.pdf](http://www.planet-data.eu/sites/default/files/Big_Data_Tutorial_part4.pdf)
3. [www.ibm.com/developerworks/data](http://www.ibm.com/developerworks/data)
4. [www.solacesystems.com](http://www.solacesystems.com)
5. [en.wikipedia.org/wiki/Big\\_data](http://en.wikipedia.org/wiki/Big_data)
6. [www.sap.com/solution/big-data.html](http://www.sap.com/solution/big-data.html)



**18ITU603B****SYSTEM PROGRAMMING****Semester – VI  
3H – 3C**

**Instruction Hours / week: L: 3 T: 0 P: 0    Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours

**Course Objectives (CO)**

- To introduce students the concepts and principles of system programming
- To provide students the knowledge about both theoretical and practical aspects of system programming, teaching them the methods and techniques for designing and implementing system-level programs.
- To train students in developing skills for writing system software with the aid of sophisticated OS services, programming languages and utility tools.
- To understand the role and functioning of various system programs over application program.
- To understand the need to follow the syntax in writing an application program and to learn the how the analysis phase of compiler is designed to understand the programmer's requirements without ambiguity.
- To synthesize the analysis phase outcomes to produce the object code that is efficient in terms of space and execution time.

**Course Outcomes (COs)**

Upon completion of the subject, students will be able to

1. Organize the functionalities and components of a computer system into different layers, and have a good understanding of the role of system programming and the scope of duties and tasks of a system programmer
2. Grasp the concepts and principles, and be familiar with the approaches and methods of developing system-level software (e.g., compiler, and networking software)
3. Apply the knowledge and techniques learnt to develop solutions to realworld problems
4. Select and make use of the OS kernel functions and their APIs, standard programming languages, and utility tools
5. Organize and manage software built for deployment and demonstration
6. Analyze requirements and solve problems using systematic planning and development approaches

**Unit I**

**Assemblers & Loaders, Linkers:** One pass and two pass assembler design of an assembler, Absolute loader, relocation and linking concepts, relocating loader and Dynamic Linking., overview of compilation, Phases of a compiler.

**Unit II****Lexical Analysis:**

Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, lexical

**Unit III****Parsing:**

Bottom up parsing- LR parser, yaITU. **Intermediate representations: Three** address code generation, syntax directed translation, translation of types, control Statements.

**Unit IV**

**Storage organization:** Activation records stack allocation.

**Unit V**

**Code Generation:** Object code generation

**Suggested Readings**

1. Santanu Chattopadhyaya. (2011). Systems Programming. New Delhi: PHI.
2. Alfred, V. Aho., Monica, S. Lam., Ravi Sethi., & Jeffrey, D. Ullman. (2006). Compilers: Principles, Techniques, and Tools (2nd ed.). New Delhi: Prentice Hall.
3. Dhamdhare, D. M. (2011). Systems Programming. New Delhi: Tata McGraw Hill.
4. Leland Beck., & Manjula, D. (2008). System Software: An Introduction to System Programming (3<sup>rd</sup> ed.). New Delhi: Pearson Education.
5. Grune, D., Van Reeuwijk, K., Bal, H. E., Jacobs, C. J. H., & Langendoen, K.(2012). Modern Compiler Design (2nd ed.). Springer.

**Semester – V****18ITU611A****PHP PROGRAMMING - PRACTICAL****4H –2C**

**Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To understand how server-side programming works on the web.
- To learn PHP Basic syntax for variable types and calculations.
- To use PHP built-in functions and creating custom functions
- To understand POST and GET in form submission.
- To receive and process form submission data.
- To create a database in phpMyAdmin, to read and process data in a MySQL database

**Course Outcomes (COs)**

Upon completion of this course, the students will be able to

1. Write PHP scripts to handle HTML forms.
2. Write regular expressions including modifiers, operators, and metacharacters.
3. Create PHP programs that use various PHP library functions, and that manipulate files and directories.
4. Analyze and solve various database tasks using the PHP language.
5. Analyze and solve common Web application tasks by writing PHP programs
6. Get hands on experience on various techniques of web development and will be able to design and develop a complete website.

**List of Programs**

1. Create a PHP page using functions for comparing three integers and print the largest number.
2. Write a function to calculate the factorial of a number (non-negative integer). The function accept the number as an argument.
3. WAP to check whether the given number is prime or not.
4. Create a PHP page which accepts string from user. After submission that page displays the reverse of provided string.
5. Write a PHP function that checks if a string is all lower case.
6. Write a PHP script that checks whether a passed string is palindrome or not? ( A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run)
7. WAP to sort an array.
8. Write a PHP script that removes the whitespaces from a string.

Sample string : 'The quick brown fox' Expected Output : Thequickbrownfox

9. Write a PHP script that finds out the sum of first n odd numbers.

10. Create a login page having user name and password. On clicking submit, a welcome message should be displayed if the user is already registered (i.e.name is present in the database) otherwise error message should be displayed.

11. Write a PHP script that checks if a string contains another string.

12. Create a simple 'birthday countdown' script, the script will count the number of days between current day and birth day.

13. Create a script to construct the following pattern, using nested for loop.

```
*
* *
* * *
* * * *
* * * * *
```

14. Write a simple PHP program to check that emails are valid.

15. WAP to print first n even numbers.

16. \$color = array('white', 'green', 'red')

Write a PHP script which will display the colors in the following way : Output :  
white, green, red,

- green • red
- white

17. Using switch case and dropdown list display a —Hello! message depending on the language selected in drop down list.

18. Write a PHP program to print Fibonacci series using recursion.

19. Write a PHP script to replace the first 'the' of the following string with 'That'.

**Sample :** 'the quick brown fox jumps over the lazy dog.'

**Expected Result :** That quick brown fox jumps over the lazy dog.

### Suggested Readings

1. Steven Holzner. (2007). PHP: The Complete Reference. New Delhi: McGraw Hill Education (India).

2. Timothy Boronczyk., & Martin, E. Psinas. (2008). PHP and MYSQL (Create-Modify-Reuse). New Delhi: Wiley India Private Limited.
3. Robin Nixon. (2014). Learning PHP, MySQL, JavaScript, CSS & HTML5 (3rd ed.). O'reilly.
4. Luke Welling.,& Laura Thompson.(2008). PHP and MySQL Web Development (4th ed.). Addition Paperback, Addison-Wesley Professional.
5. David Sklar., & Adam Trachtenberg. PHP Cookbook: Solutions & Examples for PHP.

**Web Sites**

1. <http://www.w3schools.com/php/>

---

**18ITU611B      UNIX / LINUX PROGRAMMING - PRACTICAL      Semester – VI  
4H – 2C**

---

**Instruction Hours / week: L: 0 T: 0 P: 4      Marks: Internal : 40    External : 60    Total: 100**  
End Semester Exam : 3 Hours

**Course Objectives (CO)**

- To understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.
- To understand how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions
- To understand how the operating system abstractions can be implemented
- To understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software
- To understand basic resource management techniques (scheduling or time management, space management) and principles and how they can be implemented.
- These also include issues of performance and fairness, avoiding deadlocks, as well as security and protection.

**Course Outcomes (COs)**

Upon completion of this course, the students will be able to

1. Develop software for Linux/UNIX systems.
2. Learn the C language and get experience programming in C.
3. Learn the important Linux/UNIX library functions and system calls.
4. Understand the inner workings of UNIX-like operating systems.
5. Obtain a foundation for an advanced course in operating systems.
6. Construct various shell scripts for simple applications

**List of Programs**

1. Write a shell script to check if the number entered at the command line is prime or not.
2. Write a shell script to modify —call command to display calendars of the specified months.
3. Write a shell script to modify —call command to display calendars of the specified range of months.
4. Write a shell script to accept a login name. If not a valid login name display message – —Entered login name is invalid.
5. Write a shell script to display date in the mm/dd/yy format.
6. Write a shell script to display on the screen sorted output of —who command along with the total number of users.
7. Write a shell script to display the multiplication table any number,
8. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
9. Write a shell script to find the sum of digits of a given number.
10. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.

11. Write a shell script to find the LCD (least common divisor) of two numbers.
12. Write a shell script to perform the tasks of basic calculator.
13. Write a shell script to find the power of a given number.
14. Write a shell script to find the binomial coefficient  $C(n, x)$ .
15. Write a shell script to find the permutation  $P(n, x)$ .
16. Write a shell script to find the greatest number among the three numbers.
17. Write a shell script to find the factorial of a given number.
18. Write a shell script to check whether the number is Armstrong or not.
19. Write a shell script to check whether the file have all the permissions or not

### **Suggested Readings**

1. Sumitabha, Das.(2006). Unix Concepts And Applications. New Delhi: Tata McGraw-Hill Education.
2. Michael Jang. (2011). RHCSA/ RHCE Red Hat Linux Certification: Exams (Ex200 & Ex300). Certification Press.
3. Nemeth Synder., & Hein.(2010). Linux Administration Handbook (2nd ed.). Pearson Education.
4. Richard Stevens, W. Bill Fenner., & Andrew, M. Rudoff. (2014). Unix Network Programming, The sockets Networking API, Vol. 1, (3rd ed.).

---

<b>18ITU612A</b>	<b>E-COMMERCE TECHNOLOGIES- PRACTICAL</b>	<b>Semester – VI</b>
		<b>4H – 2C</b>

---

**Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

**Course Objectives (CO)**

- To introduce the concepts, vocabulary, and procedures associated with E-Commerce and the Internet.
- To gain an overview of all aspects of E-Commerce.
- To develop the Internet and E-Commerce, options available for doing business on the Internet, features of Web sites and the tools used to build an E-Commerce web site, marketing issues, payment options, security issues, and customer service.
- It is designed to familiarize students with organizational and managerial foundations of systems, the technical foundation for understanding information systems
- Assess e-commerce strategies and applications, including online marketing, e-government, e-learning and global e-commerce

**Course Outcomes (COs)**

Upon successful completion of this course, the student will be able to:

1. Describe an example of system architecture for an e-Business.
2. List the seven major elements of web design.
3. Identify and explain fundamental web site tools including design tools, programming tools, and data processing tools.
4. Identify the major electronic payment issues and options.
5. Discuss security issues and explain procedures used to protect against security threats.
6. Identify and discuss management issues underlying e-Commerce issues including organizational structure, strategic planning, goal setting, corporate social responsibility, international arena, changing market intermediaries, resource allocation and customer service.

**List of Programs**

1. Write a HTML program to implement the use of Image map.
2. Write a CSS to implement selectors in HTML
3. Write a CSS to implement pseudo – classes with in-line styles
4. Write a Javascript program to validate a web form
5. Write a Javascript program to allow visitors to see history of visiting your page
6. Write a Javascript program to change random color each 5 seconds
7. Write a perl program to read a list of n strings (from STDIN) into an array and print a random string from the list (Use srand;rand(@array))



8. Write a perl program to read a list of n numeric's from STDIN and find the max, min, range, median and mode. Input size of the list n interactively.
9. Write a perl program to read a file of words and replaces all words in the file with their uppercase equivalent (hint: use tr/a-z/A-Z/)
10. Write VBScript program to print Fibonacci series using Do..while loop and For loop.
11. Write VBScript program to generate date and time in defferent format
12. Write VBScript program to print student marklist
13. Develop an ASP code to retrieve information from forms
14. Develop an ASP code to reading and writing cookies information
15. Develop an ASP code using response object methods

**Suggested Readings**

- 1.G.S.V.Murthy (2011). E-Commerce concepts, Models, Strategies. Himalaya Publishing house.
2. Gray. P. Schneider (2011). Electronic commerce International student edition.
3. Henry Cahn, Raymond Lee, Tharam Dillon, Elizabeth Chang. (2011). E-Commerce fundamentals and Applications. Wiley Student Edition.
4. Kamlesh K.Bajaj and Debjani Nag (2005).E-Commerce.
5. David Whitley (2000).E-Commerce-strategies, Technologies and Applications. TMH.

**Websites**

1. [http://www.tutorialspoint.com/e\\_commerce/e\\_commerce\\_tutorial.pdf](http://www.tutorialspoint.com/e_commerce/e_commerce_tutorial.pdf)
2. <http://www.dynamicwebs.com.au/tutorials/e-commerce.htm>
3. <http://www.htmlgoodies.com/beyond/webmaster/projects/electronic-commerce-tutorial.html>

**Semester – VI****18ITU612B****CLOUD COMPUTING -PRACTICAL****4H – 2C**

---

**Instruction Hours / week: L: 0 T: 0 P: 4    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

---

**Course Objectives (CO)**

- To learn how to use Cloud Services.
- To implement Virtualization
- To implement Task Scheduling algorithms.
- To apply Map-Reduce concept to applications.
- To build Private Cloud.
- To know the impact of engineering on legal and societal issues involved

**Course Outcomes (COs)**

Upon completion of this course, the students will be able to

1. Analyze the Cloud computing setup with it's vulnerabilities and applications using different architectures.
2. Design different workflows according to requirements and apply map reduce programming model.
3. Apply and design suitable Virtualization concept, Cloud Resource Management and design scheduling algorithms.
4. Create combinatorial auctions for cloud resources and design scheduling algorithms for computing clouds
5. Assess cloud Storage systems and Cloud security, the risks involved, its impact and develop cloud application
6. Broadly educate to know the impact of engineering on legal and societal issues involved in addressing the security issues of cloud computing.

**List of Programs**

1. Create virtual machines that access different programs on same platform.
2. Create virtual machines that aITUess different programs on different platforms.
3. Working on tools used in cloud computing online-
  - a) Storage
  - b) Sharing of data
  - c) Manage your calendar, to-do lists,
  - d) A document editing tool
4. Exploring Google cloud
5. Exploring Microsoft cloud
6. Exploring Amazon cloud

**Suggested Readings**

1. Barrie Sosinsky. (2010). Cloud Computing Bible. New Delhi: Wiley-India,
2. Rajkumar Buyya., James Broberg., & Andrzej, M. Goscinski Wile. Cloud Computing: Principles and Paradigms.
3. Nikos Antonopoulos., & Lee Gillam. (2012). Cloud Computing: Principles, Systems and Applications. Springer.
4. Ronald, L. Krutz., & Russell Dean Vines. (2010). Cloud Security: A Comprehensive Guide to Secure Cloud Computing. New Delhi: Wiley-India.
5. Gautam Shroff. (2010). Enterprise Cloud Computing Technology Architecture Applications. Adobe Reader ebooks available from eBooks.com.
6. Toby Velte., Anthony Velte., & Robert Elsenpeter.(2010). Cloud Computing, A Practical Approach. McGraw Hills.
7. Dimitris, N. Chorafas. (2010). Cloud Computing Strategies. CRC Press.

**Web Sites**

1. [en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing)
2. [www.ibm.com/cloud-computing/in/en/](http://www.ibm.com/cloud-computing/in/en/)
3. [www.oracle.com/CloudComputing](http://www.oracle.com/CloudComputing)
4. [www.microsoft.com/en-us/cloud/default.aspx](http://www.microsoft.com/en-us/cloud/default.aspx)

**Semester – VI****18ITU613A****BIGDATA ANALYTICS - PRACTICAL****3H – 1C**

---

**Instruction Hours / week: L: 0 T: 0 P: 3    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours**

---

**Course Objectives (CO)**

- To provide an overview of an exciting growing field of big data analytics.
- To impart to students the skills required to design scalable systems that can accept, store, and analyze large volumes of unstructured data.
- The objective of this course is to ascertain that the students know the fundamental techniques and tools used to design and analyze large volumes of data.
- To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability.
- To enable students to have skills that will help them to solve complex real-world problems in for decision support.
- To understand, and practice big data analytics and machine learning approaches

**Course Outcomes (COs)**

Upon completion of this course, the students will be able to

1. Explain the motivation for big data systems and identify the main sources of Big Data in the real world.
2. Demonstrate an ability to use frameworks like Hadoop, NOSQL to efficiently store retrieve and process Big Data for Analytics.
3. Implement several Data Intensive tasks using the Map Reduce Paradigm
4. Understand the key issues in big data management and its associated applications in intelligent business and scientific computing.
5. Interpret business models and scientific computing paradigms, and apply software tools for big data analytics.
6. Achieve adequate perspectives of big data analytics in various applications like recommender systems, social media applications etc.

**List of Programs**

1. Implement a quicksort using scala.
2. Implement a auction service using scala.
3. Write a scala function to perform any 10 arithmetic operations.
4. Write a program to find the factorial of a given number using recursion.
5. Write a program for string manipulations.
6. Write a program for alphabetic order arrangement of a set of names.
7. Write a program for student records using scala list.
8. Implement any 5 map methods for maintaining customer details.
9. Implement employee records using Files

10. Write a program to copy the files using command line arguments.

**Suggested readings**

1. Judith Hurwitz, Alan Nugent, Fern Halper, Marcia Kaufman, (2013). Big Data For Dummies, Wiley India, New Delhi.
2. Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan, (2012). Harness the Power of Big Data The IBM Big Data Platform, Tata McGraw Hill Publications, New Delhi.
3. Michael Minelli, Michele Chambers, Ambiga Dhiraj, (2013). Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses, Wiley Publications, New Delhi.
4. Zikopoulos, Paul, Chris Eaton, (2011). Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, Tata McGraw Hill Publications, New Delhi.

**Websites**

1. [www.oracle.com/BigData](http://www.oracle.com/BigData)
2. [www.planet-data.eu/sites/default/files/Big\\_Data\\_Tutorial\\_part4.pdf](http://www.planet-data.eu/sites/default/files/Big_Data_Tutorial_part4.pdf)
3. [www.ibm.com/developerworks/data](http://www.ibm.com/developerworks/data)
4. [www.solacesystems.com](http://www.solacesystems.com)
5. [en.wikipedia.org/wiki/Big\\_data](http://en.wikipedia.org/wiki/Big_data)
6. [www.sap.com/solution/big-data.html](http://www.sap.com/solution/big-data.html)

**18ITU613B****SYSTEM PROGRAMMING - PRACTICAL****Semester – VI  
3H – 1C****Instruction Hours / week: L: 0 T: 0 P: 3    Marks: Internal : 40    External : 60    Total: 100**  
**End Semester Exam : 3 Hours****Course Objectives (CO)**

- To introduce students the concepts and principles of system programming
- To provide students the knowledge about both theoretical and practical aspects of system programming, teaching them the methods and techniques for designing and implementing system-level programs.
- To train students in developing skills for writing system software with the aid of sophisticated OS services, programming languages and utility tools.
- To understand the role and functioning of various system programs over application program.
- To understand the need to follow the syntax in writing an application program and to learn the how the analysis phase of compiler is designed to understand the programmer's requirements without ambiguity.
- To synthesize the analysis phase outcomes to produce the object code that is efficient in terms of space and execution time.

**Course Outcomes (COs)**

Upon completion of the subject, students will be able to

1. Organize the functionalities and components of a computer system into different layers, and have a good understanding of the role of system programming and the scope of duties and tasks of a system programmer
2. Grasp the concepts and principles, and be familiar with the approaches and methods of developing system-level software (e.g., compiler, and networking software)
3. Apply the knowledge and techniques learnt to develop solutions to realworld problems
4. Select and make use of the OS kernel functions and their APIs, standard programming languages, and utility tools
5. Organize and manage software built for deployment and demonstration
6. Analyze requirements and solve problems using systematic planning and development approaches

**List of Programs**

1. Write a program to create a text editor.
2. Write a program to implement an absolute loader.
3. Write a program to check balance parenthesis of a given program.
4. Write a program to check the valid or invalid identifier
5. Write a program to implement DFA
6. Write a program to remove blank space in a given string.
7. Write a program to identify tokens in a given string.

8. Write a program to remove special character in a given string.
9. Write a program to check given string is a keyword or not.
10. Write a program to identify tokens in a given expression.

**Suggested Readings**

1. Santanu Chattopadhyaya. (2011). Systems Programming. New Delhi: PHI.
2. Alfred, V. Aho., Monica, S. Lam., Ravi Sethi., & Jeffrey, D. Ullman. (2006). Compilers: Principles, Techniques, and Tools (2nd ed.). New Delhi: Prentice Hall.
3. Dhamdhere, D. M. (2011). Systems Programming. New Delhi: Tata McGraw Hill.
4. Leland Beck., & Manjula, D. (2008). System Software: An Introduction to System Programming (3<sup>rd</sup> ed.). New Delhi: Pearson Education.
5. Grune, D., Van Reeuwijk, K., Bal, H. E., Jacobs, C. J. H., & Langendoen, K.(2012). Modern Compiler Design (2nd ed.). Springer.