கற்பகம்உயர்கல்விகலைக்கழகம்

தமிழ்த்துறை

பகுதி - I தமிழ்ப்பாடத்திட்டம் (2017 - 2018)

இரண்டாம்பருவம்

(இளநிலை அறிவியல் பட்டவகுப்புகளுக்குரியது)

(For I-UG Science Degree Classes)   17LSU101

பாடத்திட்டப்பொதுநோக்கம்

- கற்றல் வழி சிந்தனைத் திறனையும், கருத்து வெளிப்பாட்டுத் திறனையும், மேம்படுத்துதல்.
- ஆய்வுநோக்கைமேம்படுத்துதல்.
- 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.

பாடத்திட்டப்பயன்விளைவு

- இந்தியகுடியுரிமைப்பணிமுதலானபோட்டித்தேர்வுகளில், விருப்பப்பாடமாகஇடம்பெறுகின்ற, ‘தமிழ்இலக்கியவரலாறு’ குறித்தமுழுமையானஅறிமுகம்பெற்றிருத்தல்.
- கல்வெட்டியல், ஓலைச்சுவடியியல்மற்றும்தொல்லியல்சார்ந்தஆவணத்தேடலுக்குரியஆய்வுமனப்பான்மையுடன், இலக்கியங்களைஅணுகுதல்.
- தமிழின்வளர்ச்சித்துறையாகிய, ‘அறிவியல்தமிழ்’ ; ‘இணையதமிழ்’ குறித்தபன்நோக்குஅணுகுமுறையிலானஆய்வுச்சிந்தனைமேம்பாடு.
- வேலைவாய்ப்புக்குரியசுயதிறன்மேம்பாட்டுடன், படைப்பாக்கத்திறன்மேம்பாடும்பெற்றிருத்தல் .
- சமுதாயமற்றும்வாழ்வியல்மதிப்புகளைப்பேணுவதற்குக்கருவியாகஇலக்கியங்களைநாடுகின்றமனப்பான்மைவளர்ச்சி.
- மொழிபெயப்புத்துறைசார்ந்தவேலைவாய்புத்திறன்பெற்றிருத்தல்

.

தாள்கள்வரிசையும்தேர்வுச்செயல்திட்டமும்பகுதி-I தமிழ்

பருவம்	தாள்	கற்பிக்கும் நேரம்/வாரம்	தேர்வு மணிகள்	மதிப்பெண் அக/எழுத்து	மொத்தம்	மதிப்பீடு
இரண்டு	II	4	3	40 / 60	100	4

பகுதி – I, தமிழ் **பருவம் I**

17LSU101 : தமிழ் முதல் தாள் 4-H,4-C

(இளநிலை அறிவியல் பட்ட வகுப்புகளுக்குரியது)

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

கல்வி : மகாகவி பாரதியார் – சுயசரிதை - ஆங்கிலக் கல்வி.

இன்றைய நிலை : கவிமணி தேசிக விநாயகம் பிள்ளை – ஒற்றுமையே உயிர்நிலை.

மனிதநேயம் : கவிஞர் சிற்பி பாலசுப்பிரமணியன் –மலையாளக் காற்று.

சூழலியல் : கவிஞர் வைதீஸ்வரன் - விரல் மீட்டிய மழை.

பெண்ணியம் : கவிஞர் சுகந்தி சுப்பிரமணியம் – புதையுண்ட வாழ்க்கை.

அலகு – II : அற இலக்கியம்: **(10 மணிநேரம்)**

கொன்றை வேந்தன்: 1-50 பாடல்கள்

திருக்குறள்: பண்புடைமை, வினைத்திட்பம் – 20 குறள்கள்

பழமொழி நானூறு: 5 பாடல்கள்

அலகு - III : சிற்றிலக்கியம்: **(10 மணிநேரம்)**

மூவருலா: 1-26 கண்ணிகள்

திருச்செந்தூர் முருகன் பிள்ளைத்தமிழ்: 2 பாடல்கள்

கலிங்கத்துப் பரணி: போர்பாடியது - 9 பாடல்கள்

அலகு – IV : கட்டுரை: **(10 மணிநேரம்)**

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

அலகு- V : மொழிப்பயிற்சி: **(8 மணிநேரம்)**

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

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

Semester – I

17CAU101 PROGRAMMING FUNDAMENTALS USING C / C++ 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Int.: **40** Ext.: **60** Total: **100**

Course Objectives

Enable the student

- 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 study the concepts of Call by value and call be reference.
- To learn 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.

Course Outcomes (COs)

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

1. Understand data types and control structures in programming languages.
2. Solve the memory access problems by using pointers
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. Apply pointer concepts in C++
6. Use the concepts of preprocessor directives and macros.

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, Scope 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 etc), 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

Semester – I

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. Andrew Koenig, Barbara, E. Moo, (2000). *ACCelerated C++*. Addison-Wesley.
2. Balaguruswamy,E.,(2008). *Object Oriented Programming with C++*. Tata McGraw-Hill Education.
3. Bjarne Stroustroup, (2014). *Programming -- Principles and Practice using C++*. (2nd ed.). Addison-Wesley.
4. Bjarne Stroustrup, (2013). *The C++ Programming Language*, (4th ed.). Addison-Wesley.
5. Harry, H. Chaudhary,(2014). *Head First C++ Programming*: *The Definitive Beginner's Guide*. CreateSpace Independent Publishing Platform.
6. Herbtz Schildt, (2012). *C++: The Complete Reference*. (5th ed.). McGraw-Hill Osborne Media
7. John R. Hubbard, (2000). *Programming with C++*. (2nd ed.). Schaum's Series.
8. Paul Deitel, Harvey Deitel, (2011).*C++ How to Program*. (8th ed.). Prentice Hall.
9. Scott Meyers, (2005). *Effective C++*, (3rd ed.), Addison-Wesley.
10. Stanley B. Lippman, JoseeLajoie, Barbara E. Moo,(2012). *C++ Primer*. (5th ed.). Addison-Wesley.
11. Walter Savitch, (2007). *Problem Solving with C+*., (7th ed.). Pearson Education.

Websites

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/
6. www.cppreference.com/

Semester – I

17CAU102 COMPUTER SYSTEM ARCHITECTURE 4H – 4C

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

Course Objectives

Enable the student

- To discuss the basic concepts of Logic gates, multiplexers and de multiplexers.
- 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.
- Explain different types of addressing modes.
- To learn CISC and RISC architecture.
- To summarize the types of memory and I/O channels.

Course Outcomes (COs)

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

1. Students will acquire a basic knowledge about computer system architecture, arithmetic, digital circuits and the low - level programming skills.
2. Define different number systems, binary addition and subtraction, 2's complement representation and operations with this representation.
3. Understand the inner workings and performance capabilities of microprocessors.
4. Understand the concept of registers and Instruction set.
5. Know the addressing modes and instruction format of various microprocessors.
6. Understand memory and I/O channels.

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

Semester – I

17CAU103 INTRODUCTION TO INFORMATION TECHNOLOGY 4H – 4C

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

Course Objectives

Enable the student

- 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)

Upon successful completion of the course, the 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,
4. Gain knowledge about five generations of computer system,
5. Explain the functions of a computer,
6. Identify and discuss the functional units of a computer system,

Unit- I

Introduction to Computers-Classification of Digital Computer systems-Anatomy of a digital computer-Computer architecture - Data and Information-Introduction - Types of data -Data processing using a computer - Desktop computer - Acquisition of numbers and textual data-Introduction-input units-Internal representation of numeric data-representation of characters in Computers-Error detecting codes.

Unit-II

Acquiring graphical data-Introduction-Acquisition of textual data and pictures-Storage format for pictures-Image compression fundamentals-Image acquisition with digital

Semester – I

17CAU103 INTRODUCTION TO INFORMATION TECHNOLOGY 4H – 4C

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

camera-Acquiring audio data-Acquisition of video processing textual and multimedia data.

Unit-III

Data storage-Memory cell-physical devices used as memory cells-RAM-ROM-Secondary memory-I/O devices-CPU-Data Organization.

Unit-IV

Computer software and Software Development: Introduction to Computer Software-Operating systems-Functions of an OS-Classification of OS-Programming Languages-Compilers and Interpreters-General software features and trends.

Unit-V

Computers and Communication-Internet Applications-Introduction-E-mail-Information browsing service-world wide web-Information retrieval from worldwide web-other facilities provided by browser-Audio-Graphics-Animation and Video on the Internet-Business Information Systems.

Suggested readings

1. Alexis Leon and Mathews Leon. (1999). “***Fundamentals of Information Technology***.” (1st ed.).Leon Tech World.
2. Rajaraman,V.,(2003). “***Introduction to Information Technology***”. (1st ed.). Prentice Hall of India. Delhi.
3. Anurag Jain,(2002). “***Information Technology***”. (1st ed.) Excel Books.

Websites

1. www.tesu.edu/watson/IT
2. www.bcanotes.com/It.html
3. ww.amazon.in/Fundamentals of IT

Semester – I

17CAU111 PROGRAMMING FUNDAMENTALS USING C / C++ PRACTICAL **3H – 2C**

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

Course Objectives

Enable the student

- 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 study the concepts of Call by value and call be reference.
- To learn 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.

Course Outcomes (COs)

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

1. Develop programs using the basic elements like control statements, Arrays and Strings .
2. Solve the memory access problems by using pointers
3. Learn the basics of file handling mechanism that is essential for understanding the concepts in database management systems.
4. Understand the uses of preprocessors and various header file directives.
5. Use the characteristics of an object-oriented programming language in a program.
6. Implement File handling mechanisms.

List of Programs:

1. Write a program to print the sum and product of digits of an integer.

2. Write a program to reverse a number.

3. Write a program to compute the sum of the first n terms of the following series S = 1+1/2+...

4. Write a program to compute the sum of the first n terms of the following series S=1-2+3-4+5..

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. Write a program to compute the factors of a given number.

17CAU111 PROGRAMMING FUNDAMENTALS USING C / C++ PRACTICAL 3H – 2C

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

8. Write a macro that swaps two numbers. Write a program to use it.

9. Write a program to print a triangle of stars as follows (take number of lines from user):

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

10. Write a program 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

11. Write a program that swaps two numbers using pointers.

12. Write a program in which a function is passed address of two variables and then alter its contents.

13. 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.

14. 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.

15. Write a program to calculate Factorial of a number (i)using recursion, (ii) using iteration

Semester – I

17CAU112 COMPUTER SYSTEM ARCHITECTURE (PRACTICAL) 3H – 2C

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

(Any 8 Experiments)

Course Objectives

Enable the student

- To discuss the basic concepts of Logic gates, multiplexers and de multiplexers.
- 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.
- Explain different types of addressing modes.
- To learn CISC and RISC architecture.
- To summarize the types of memory and I/O channels.

Course Outcomes (COs)

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

1. Students will acquire a basic knowledge about computer system architecture, arithmetic, digital circuits and the low - level programming skills.
2. Understand the inner workings and performance capabilities of advanced microprocessors.
3. analyze the use of encoder and decoder.
4. Understand the working of half adder and full adder.
5. Design and analyze memory, I/O devices and cache structures for processor.
6. Evaluate the performance of computer systems

List of Programs:

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

Semester – I

17CAU113 INTRODUCTION TO INFROMATION TECHNOLOGY PRACTICAL 4H – 2C

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

Course Objectives
Enable the student

- Create a document in Microsoft Word with formatting options.
- Add a graphic to a document.
- Write functions in Microsoft Excel to perform basic calculations and to convert number to text and text to number.
- Indicate the names and functions of the Excel interface components.
- Construct formulas, including the use of built-in functions, and relative and absolute references.
- Create and modify charts.

Course Outcomes (COs)
Upon successful completion of the course, the student will be able to:

1. Modify text using various formatting options from the editing tools under the Home tab
2. Demonstrate the mechanics and uses of Word tables to organize and present data.
3. Demonstrate working knowledge of using Word's themes and clip art to create a variety of visual effects.
4. Create and design a spreadsheet for general office use.
5. Demonstrate the basic mechanics and navigation of an Excel spreadsheet.
6. Demonstrate formatting techniques and presentation styles.

List of Programs:

1. Create a set of slides with sound and animation using power point and to select various styles of slides from slide template.
2. Prepare an organization chart for a company and a college using power point.
3. To perform comparative study of your UG degree subject marks and create a various styles of graph using excel.
4. To perform student and hostel fee particulars in a worksheet. Analysis the results using excel.
5. To create a word document for salary preparation and access the calculation, graph from excel using OLE.
6. Apply the mail merge concept using word.
7. Image compression using Java.
8. Animation using Java.
9. Create simple accounts using Tally.
10. Create balance sheet using Tally.

Semester – I

17AEC101 ENVIRONMENTAL SCIENCE 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 Marks: Internal: **100** Total: **100**

Course Objectives

Enable the student

- 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 encourage to learn ecosystems and biodiversity.
- To learn environment pollution and control measures of pollution.
- To create system concepts and methodologies and analyze interactions.

Course Outcomes (COs)

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

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

Environment Definition, scope and importance, components, Ecosystem Definition, Concept, Scope, 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.

Semester – I

17AEC101 ENVIRONMENTAL SCIENCE 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **100** Total: **100**

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. Tripathy.S.N. and Sunakar Panda. (2004). Fundamentals of Environmental Studies; 2nd Edition, Vrianda Publications Private Ltd., New Delhi.
2. Arvind Kumar,(2004). A Textbook of Environmental Science; APH Publishing Corporation, New Delhi.
3. Verma, P.S., Agarwal, V.K.,(2001). Environmental Biology (Principles of Ecology); S.Chand and Company Ltd., New Delhi.

Semester – I

17AEC101 ENVIRONMENTAL SCIENCE 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **100** Total: **100**

4. Anubha Kaushik, Kaushik, C.P. ,(2004). Perspectives in Environmental Studies, New Age International Pvt. Ltd. Publications, New Delhi.
5. Singh, M.P., B.S. Singh and Soma S. Dey, (2004). Conservation of Biodiversity and Natural Resources. Daya Publishing House, Delhi.
6. Daniel B.Botkin and Edward A. Keller, (1995). Environmental Science, John Wiley and Sons, Inc., New York.
7. Uberoi, N.K., (2005). Environmental Studies, Excel Books Publications, New Delhi, India.

கற்பகம்உயர்கல்விகலைக்கழகம்

தமிழ்த்துறை

பகுதி - I தமிழ்ப்பாடத்திட்டம் (2017 - 2018)

இரண்டாம்பருவம்

(இளநிலை அறிவியல் பட்டவகுப்புகளுக்குரியது)

(For I-UG Science Degree Classes) 17LSU201

பாடத்திட்டப்பொதுநோக்கம்

- கற்றல் வழி சிந்தனைத் திறனையும், கருத்து வெளிப்பாட்டுத் திறனையும், மேம்படுத்துதல்.
- ஆய்வுநோக்கைமேம்படுத்துதல்.
- 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.

பாடத்திட்டப்பயன்விளைவு

- இந்தியகுடியுரிமைப்பணிமுதலானபோட்டித்தேர்வுகளில், விருப்பப்பாடமாகஇடம்பெறுகின்ற, ‘தமிழ்இலக்கியவரலாறு’ குறித்தமுழுமையானஅறிமுகம்பெற்றிருத்தல்.
- கல்வெட்டியல், ஓலைச்சுவடியியல்மற்றும்தொல்லியல்சார்ந்தஆவணத்தேடலுக்குரியஆய்வுமனப்பான்மை யுடன், இலக்கியங்களைஅணுகுதல்.
- தமிழின்வளர்ச்சித்துறையாகிய, ‘அறிவியல்தமிழ்’ ; ‘இணையதமிழ்’ குறித்தபன்நோக்குஅணுகுமுறையிலானஆய்வுச்சிந்தனைமேம்பாடு.
- வேலைவாய்ப்புக்குரியசுயதிறன்மேம்பாட்டுடன், படைப்பாக்கத்திறன்மேம்பாடும்பெற்றிருத்தல் .
- சமுதாயமற்றும்வாழ்வியல்மதிப்புகளைப்பேணுவதற்குக்கருவியாகஇலக்கியங்களைநாடுகின்றமனப்பான்மைவளர்ச்சி.
- மொழிபெயப்புத்துறைசார்ந்தவேலைவாய்புத்திறன்பெற்றிருத்தல்

.

தாள்கள்வரிசையும்தேர்வுச்செயல்திட்டமும்பகுதி-I தமிழ்

பருவம்	தாள்	கற்பிக்கும் நேரம்/வாரம்	தேர்வு மணிகள்	மதிப்பெண் அக/எழுத்து	மொத்தம்	மதிப்பீடு
இரண்டு	II	4	3	40 / 60	100	4

பகுதி – I, தமிழ் **பருவம் II**

17LSU201 : 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 : சுடர்தொடீ 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: 'என்னே' 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.

Semester – II

17CAU201 PROGRAMMING IN JAVA 4H – 4C

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

Course Objectives

- 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)

1. Student will obtain knowledge of the structure and model of the Java programming language.
2. How to 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. propose the use of 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, Scope, 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, Interfaces, Packages, Enumerations, Autoboxing and Metadata: 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, Threading, Networking 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 Edition.
2. James Gosling, Bill Joy, Guy L Steele Jr, GiladBracha, Alex Buckley,(2014). The Java Language Specification, Java SE, 8th edition (Java Series)", Published by Addison Wesley.
3. Joshua Bloch, (2008)."Effective Java" 2nd edition,Publisher: Addison-Wesley.
4. Cay S. Horstmann, GaryCornell,(2012). "Core Java 2 Volume 1 ,9th Edition,Printice Hall.

5. Cay S. Horstmann, Gary Cornell, (2013). "Core Java 2 Volume 2 - Advanced Features)", 9th edition, Printice Hall.

6. Bruce Eckel,(2002). "Thinking in Java", 3rd Edition, PHI.

7. Balaguruswamy,E.,(2009). "Programming with Java", 4th Edition, McGraw Hill.

8. Paul Deitel, Harvey Deitel,(2011). "Java: How to Program", 10th Edition, Prentice Hall.

9. Bert Bates, Kathy Sierra, (2005). "Head First Java", Orielly Media Inc. 2nd Edition.

10. David J. Eck, (2009). "Introduction to Programming Using Java", Published by Create Space Independent Publishing Platform.

11. John R. Hubbard,(2004). "Programming with JAVA", Schaum's Series, 2nd Edition.

Websites

1. java.sun.com/docs/books/tutorial/
2. www.en.wikipedia.org/wiki/Java
3. www.java.net/

Semester – II

17CAU202 **DISCRETE STRUCTURES** **4H – 4C**

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

Course Objectives

Enable the student

- To develop logical thinking and its application to computer science
- To write and evaluate a proof or outline the basic structure of and give examples of each proof technique described
- To know the concepts of sets and functions.
- to transform a given problem into a combination of several simpler statements and arrive at a solution to the problem and be able to prove it logically
- To solve problems with the help of tools of mathematical analysis.
- To reason and ability to present a coherent and mathematically accurate argument.

Course Outcomes (COs)

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

1. Familiar with elementary algebraic set theory.
2. Demonstrate the ability to write and evaluate a proof or outline the basic structure of and give examples of each proof technique described.
3. Understand the basic principles of sets and operations in sets.
4. Demonstrate an understanding of relations and functions and be able to determine their properties.
5. Demonstrate different traversal methods for trees and graphs.
6. Initiate to knowledge from inference theory.

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

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

Suggested readings

1. Kenneth Rosen, (2006). Discrete Mathematics and its Applications, 6th edition, McGraw Hill, New Delhi.

2. Tremblay,J .P., and Manohar, R., (1997). Discrete Mathematical Structures with Applications to Computer Science, McGraw-Hill Book Company, New Delhi.

3. Coremen, T.H., Leiserson, C.E., Rivest, R. L., (2009). Introduction to algorithms, 3rd edition, Prentice Hall on India.

4. Albertson,M. O., and Hutchinson, J. P.,(1988). Discrete Mathematics with Algorithms,
   John wiley Publication.

5. Hein, J. L., (2009). Discrete Structures, Logic, and Computability, 3rd edition, Jones and Bartlett Publishers.

6. Hunter, D.J., (2008). Essentials of Discrete Mathematics, Jones and Bartlett Publishers.

		Semester – II
17CAU203	**SCRIPT LANGUAGE**	**4H – 4C**

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

Course Objectives

Enable the student

- To study the basics of HTML
- To study the basics of Java Script
- To study the basics of Cascading style sheets and DHTML
- To study the basics of XML
- To understand the principles of creating an effective web page.
- To develop skills in analyzing the usability of a website.

Course Outcomes (COs)

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

1. Independently understand basic of web design languages.
2. To develop a website
3. Enumerate the architecture of website.
4. 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

Unit- I

HTML: Introduction - SGML - Outline of HTML Document - Head section-Body section - HTML forms.

Unit-II

Introduction - Language elements - Object of java script - Other objects - Arrays.

Unit- III

DHTML: Cascading style sheets - DHTML Document object model and collections - Event Handling - Filters and Transitions - Data binding

Unit- IV

XML: Syntax of XML Document - XML Attributes - XML Validation - XML DTD - XML -DTD Elements - DTD Attributes - DTD Entities - DTD Validation

Unit-V

PHP: Introduction to PHP - Syntax - Saving PHP files - Variables - Constants - If and Switch Statements - Operators - Loops and Strings.

Suggested readings

1. Gopalan,N.P.,.Akilandeswari,J., (2007). Web Technology: A Developer's Perspective , 2nd Edition , Prentice Hall of India. New Delhi. [Unit I – II]
2. Ivan Bayross.,(2009). Web Enabled Commercial Application Development using HTML, DHTML ,JavaScript, Perl CGI 2nd Edition, BPB Publications, New Delhi. [Unit III – V]
3. Ashok Lodha.(2007). Guide to PHP, 1st Edition. LawPoint, Kolkata.
4. Dave W.Mercer, Allan Kent, Steven D.Nowicki, Davd Mercer, Dan Squie, Wankyu Choi.(2006). Beginning PHP5, Wiley India (P) Ltd. New Delhi.
5. Tim Converse ,Joyce Park, Clark Morgan.,(2006). PHP5 & MySQL Bible, 1st Edition, John Wily, India.

Websites

1. http://www.mvps.org/scripting/languages
2. Http://en.wikipedia.org/wiki/script_language
3. http://www.mvps.org/scripting/languages

Semester – II

17CAU211 PROGRAMMING IN JAVA PRACTICAL 3H – 2C

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

Course Objectives

Enable the student

- 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 implement the Exception Handling and AWT component and AWT classes.

Course Outcomes (COs)

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

1. Student will obtain knowledge of the structure and model of the Java programming language.
2. How to 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. propose the use of 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 .lenth 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 StringBufferclasss 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 aCSUess 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 FibonaCSUi series, where the function to generate fibonacii 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 examine the different components of the URLand 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 scrolls 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.

Semester – II

17CAU212 DISCRETE STRUCTURES PRACTICAL 3H – 2C

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

Course Objectives

Enable the student

- To develop logical thinking and its application to computer science
- To write and evaluate a proof or outline the basic structure of and give examples of each proof technique described
- To know the concepts of sets and functions.
- to transform a given problem into a combination of several simpler statements and arrive at a solution to the problem and be able to prove it logically
- To solve problems with the help of tools of mathematical analysis.
- To reason and ability to present a coherent and mathematically accurate argument.

Course Outcomes(COs)

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

1. Familiar with elementary algebraic set theory.
2. Demonstrate the ability to write and evaluate a proof or outline the basic structure of and give examples of each proof technique described.
3. Understand the basic principles of sets and operations in sets.
4. Demonstrate an understanding of relations and functions and be able to determine their properties.
5. Demonstrate different traversal methods for trees and graphs.
6. Initiate to knowledge from inference theory.

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) ………+1/(n(n+1)) = n/(n+1)

4. Write a C Program to to perform the sum = 1+ (1+2) + (1+2+3) + … + (1+2…+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) $P\wedge Q\wedge \rceil R$ (ii) $P\wedge \rceil Q\wedge R$ (iii) $P\wedge Q\wedge \rceil R$

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

Semester – II

17CAU213 SCRIPT LANGUAGE PRACTICAL 4H – 2C

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

Course Objectives

Enable the student

- To understand the principles of creating an effective web page.
- To develop skills in analyzing the usability of a website.
- To learn the language of HTML and JavaScript.
- To learn to design web pages using Forms, Tables, Frames, TextBox
- To study the basics of JavaScript
- To learn to validate input in web pages using JavaScript

Course Outcomes (COs)

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

1. Gain the skills and project-based experience needed for entry into web design and development careers.
2. 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
3. Select and apply markup languages for processing, identifying, and presenting of information in web pages.
4. Create and manipulate web media objects using editing software.
5. Create web pages and validate input using JavaScript.
6. Develop web pages and perform computation using JavaScript.

List of Programs:

1. Create a form to reserve a ticket in the railways if the source and destination place is given.
2. Create a web page to display student mark statement.
3. Design an home page for a company
4. Develop DHTML Script to illustrate Color and Background attribute.
5. Create a DHTML page using various filters on images, mask image, mask Text.
6. Design a PHP program to find greatest of three numbers.
7. Design an student application form using PHP program.
8. Write a java script program design a calculator

Semester – II

17CAU213 SCRIPT LANGUAGE PRACTICAL 4H – 2C

9. Develop a Java script program for display greetings based on Time
10. Create an XML document for student information with relevant attributes and validation.
11. Develop a program to copy the content of one file to another file using PHP program.
12. Develop an E-mail application using PHP program.

Semester – II

17ENU201 **ENGLISH** **4H – 4C**

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **100** Total: **100**

Course Objectives:

- To help students enhance their Language skills
- To introduce different kinds of literary works
- To familiarize different genres of Literature
- To instruct moral values through literature.
- To improvise their productive and receptive skills
- To strengthen the basic knowledge about grammar

Course Outcome:

1. Develop the four types of skills
2. Reading and comprehending literary works
3. Genres of literature to provide moral education
4. Develop communication skills in business environment
5. Interpersonal skills will be developed.
6. Betterment of language competence

Unit-I

Prose: Google Guys (Extract) – Richard L Brandt
Poetry: The Blind Pedlar – Osbert Sitwell
Short Story: A Garden So Rich – Christie Craig
Vocabulary: Prefix, Antonyms, Sentence Completion
Grammar: Article, Adverb, Pronoun

Unit-II

Prose: Happiness 101 – Geeta Padmanabhan
Poetry: An Old Woman **–** Arun Kolatkar
Vocabulary: Suffix, Analogies
Grammar: Noun, Adjective

Unit-III

Prose: Structured Procrastination – John Perry
Short Story: The Umbrella Man – Roald Dahl
One-Act Play: The Boy Who Stopped Smiling – Ramu Ramanathan
Vocabulary: Synonyms, Euphemisms, Word Definitions
Grammar: Verb, Conjunction and Interjection, Indirect/Reported Speech

Unit-IV

Poetry: No Sentence – Anjum Hassan
One-Act Play: While the Auto Waits- O' Henry
Vocabulary: Words Often Confused, Anagrams

Grammar: Preposition, Voice- Active and Passive

Unit-V

Short Story: The Bird – Amar Jalil
One-Act Play: The Cellphone Epidemic – Claudia I. Haas
Vocabulary: Portmanteau Words, One Word Substitution
Grammar: Question, Pronunciation

Prescribed Text:

Rao, G. Chandralekha et al. (2013).*Spring* **.** Emerald Publishers: Chennai.

Suggested Reading:

Shyamala,V. (2006). *English for Communication*. 2006. Emerald Publishers: Chennai.

Semester – III

17CAU301 **DATA STRUCTURES** **4H – 4C**

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

Course Objectives:

Enable the student

- To understand the fundamental concepts of data structures
- To Learn linear data structures – lists, stacks, and queues
- To apply Tree and Graph structures
- To understand sorting, searching and hashing algorithms
- To develop application using data structures
- Choose appropriate data structures and algorithms for problem solving.

Course Outcome:

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

1. Implement abstract data types for linear data structures.
2. Apply the different linear and non-linear data structures to problem solutions.
3. Analyze the applications of tree.
4. Implement graph theory over various data structures.
5. Critically analyze the various sorting algorithms.
6. Able to use linear and non-linear data structures like stacks, queues , linked list etc.

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 collusion by Open Addressing, Coalesced Hashing, Separate Chaining, Dynamic and Extendible Hashing, Choosing a Hash Function, Perfect Hashing, Function

Suggested readings

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

Websites

1. http://en.wikipedia.org/wiki/Data_structure
2. http://www.cs.sunysb.edu/~skiena/214/lectures/
3. www.amazon.com/Teach-Yourself-Structures-Algorithms

Semester – III

17CAU302 **OPERATING SYSTEMS** **4H – 4C**

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

Course Objectives:

Enable the student

- To Study the basic concepts and functions of operating systems.
- To understand the structure and functions of OS.
- To Learn about Processes, Threads and Scheduling algorithms.
- To Understand the principles of concurrency, Deadlocks and Memory Management
- To Learn about the Protection and Security Concepts.
- Understand basic resource management techniques.

Course Outcome (COs):

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

1. Design various Scheduling algorithms.
2. Apply the principles of concurrency.
3. Design deadlock, prevention and avoidance algorithms.
4. Compare and contrast various memory management schemes.
5. Apply the Security Concepts based on Authentication.
6. Appreciate the role of operating system as System software.

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 aCSUess Authorization.

Suggested readings

1. Silberschatz,A., Galvin, P.B., Gagne,G.,(2008). *Operating Systems Concepts*, (8th ed.), John Wiley Publications.

2. Stallings,W., (2008). *Operating Systems, Internals & Design Principles*, (5th Edition), Prentice Hall of India.

3. Tanenbaum, A.S., (2007). *Modern Operating Systems*, (3rd ed.), Pearson Education.

Websites

1. www.cs.columbia.edu/~nieh/teaching/e6118_s00/
2. www.clarkson.edu/~jnm/cs644pages.cs.wisc.edu/~remzi/Classes/736/Fall2002/

Semester – III

17CAU303 COMPUTER NETWORKS 4H – 4C

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

Course Objectives:

Enable the student

- To study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model.
- To read the fundamentals and basics concepts of Physical layer with real time examples
- To study data link layer concepts, design issues, and protocols.
- To learn the functions of network layer and the various routing protocols.
- To familiarize the functions and protocols of the Transport layer and Application layer.
- Channel error detection and correction, MAC protocols, Ethernet and WLAN.

Course Outcome:

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

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

Unit- I

Introduction to Computer Networks : Network definition; network topologies; network classifications; network protocol; 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

Digital to analog modulation-; multiplexing techniques- FDM, TDM; transmission media.

Networks Switching Techniques and Access mechanisms: Circuit switching; packetswitching - 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*, (4th ed.), THM.
2. Tanenbaum, A. S. , (2002). Computer Networks**,** (4th ed.), PHI.

Websites

1. en.wikipedia.org/wiki/Internet_protocol_suite
2. http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies
3. www.yale.edu/pclt/COMM/**TCPIP**.HTM
4. www.w3schools.com/**tcpip**/default.asp
5. https://www.cse.iitb.ac.in/

Semester – III

17CAU304A **ANDROID PROGRAMMING** **3H – 3C**

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

Course Objectives:

Enable the student

- To compare the differences between Android and other mobile development environments.
- To learn the Object-oriented features of Kotlin and APIs for Android Development.
- To describe the working of Android applications, life cycle, manifest, and Intents
- To demonstrate the implementation of Form widgets for Android App development.
- To learn the SQLite database connectivity and database operations with android
- To design and develop useful Android applications with compelling user. interfaces .by using, extending, and creating your own layouts and Views and using Menus.

Course Outcome (Cos):

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

1. Analyze the Architecture and features of Android with another Mobile Operating System.
2. Evaluate the standard of Kotlin language for developing Android Applications
3. Apply knowledge for creating user Interface and develop activity for Android App.
4. Evaluate the user interface architecture of Android for developing Android Apps
5. Understand the implementation of SQLite database operations with Android.
6. Design and implement Database Application and Content providers.

Unit-I

Introduction: History of Android, Introduction to Android Operating Systems, Android Development Tools, Android Architecture. (2L)

Unit-II

Overview of object oriented programming using Java: OOPs Concepts: Inheritance, Polymorphism, Interfaces, Abstract class, Threads, Overloading and Overriding, Java Virtual Machine. (4L)

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. (5L)

Unit-IV

User Interface Architecture: Application context, intents, Activity life cycle, multiple screen size s.(2L) **User Interface Design:** Form widgets, Text Fields, Layouts, Button control, toggle buttons, Spinners(Combo boxes),Images, Menu, Dialog.(2L)

Unit-V

Database: Understanding of SQL database, connecting with the database. (2L)

Suggested readings

1. James C.Sheusi,(2013). *Android application development for Java for Java programmers*, Cengage Learning.

Websites

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://developer.android.com/guide/components/activities.html
5. http://developer.android.com/guide/components/fundamentals.html
6. http://developer.android.com/guide/components/intents-filters.html.
7. http://developer.android.com/training/multiscreen/screensizes.html
8. http://developer.android.com/guide/topics/ui/controls.html
9. http://developer.android.com/guide/topics/ui/declaring-layout.html
10. http://developer.android.com/training/basics/data-storage/databases.html

Semester – III

17CAU304B STRUTS FRAMEWORK 3H – 3C

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

Course Objectives:

Enable the students to

- Understand the Model, View, Controller (MVC) design pattern and how it is applied by Struts Framework
- Know the components of Struts Application and database connectivity
- Implement JSP functions using Struts
- Perform client and server side validation using Struts Validator Framework
- Develop web applications using Struts
- Develop EJB programs and get familiar with Struts framework

Course Outcome (COs):

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

1. Know about MVC and overview of JSP
2. Identify the components of a Struts Application and how to connect database in web based applications
3. Know about the struts Sub elements and Control Tags
4. Develop programs with Data Tags and Bean Tags
5. Develop programs with HTML Tags and Logic Tags and able to construct enterprise-level web based applications
6. Construct web based applications and Identify where data structures are appearing in them.

Unit-I

Introduction, Understanding the MVC Design Pattern, The Struts Implementation of the MVC, Directory Structure, Web Application Deployment Descriptor, The Tomcat JSP/Servlet Container, Installing and Configuring Tomcat, Testing Your Tomcat Installation, An Overview of the Java Servlet and JavaServer Pages, The GenericServlet and HttpServlet Classes, Life Cycle of a Servlet. Struts OverView, Life Cycle of Struts.

Unit-II

Components of a Struts Application, The Controller, The View, DynaActionForm & LazyDynaBean, ActionServlet, RequestProcessor, ActionForm, IncludeAction, Forward Action, LocaleAction, DispatchAction, LookupDispatchAction, MappingDispatchAction, EventDispatchAction, SwitchAction, Interceptors, Implementing Custom interceptors, Struts Validation, Exception Handling, Managing Errors, Struts Error Management - ActionError, ActionErrors, Creating Custom ActionMappings, Struts JDBC Connection, Using a DataSource in Struts Application, Debugging Struts Applications.

Unit-III

The struts-config.xml, The Struts Subelements, The icon Tag Subelement, display-name Tag Subelement, description Tag Subelement, set-property Tag Subelement, Adding a Struts DataSource, Adding FormBean Definitions, Adding Global Forwards, Adding Actions, Adding a RequestProcessor, Adding Message Resources, Adding a Plug-in. The Bean Tag Library, Installing the Bean Tags, bean:cookie Tag, bean:define Tag, bean:header Tag, bean:include Tag, bean:message Tag, bean:page Tag, bean:parameter Tag ,bean:resource Tag,<bean:size Tag, bean:struts Tag, bean:write Tag

Unit-IV

HTML Tag Library, Base Tag, Button Tag, Cancel Tag, Checkbox Tag, Errors Tag, Form Tag, Hidden Tag, Html Tag, Image Tag, Img Tag, Link Tag, Multibox Tag, Select Tag, Option Tag, Options Tag, Password Tag, Radio Tag, Reset Tag, Rewrite Tag, Submit Tag, Text Tag, Textarea Tag

Unit-V

The Logic Tag Library , Empty Tag, notEmpty Tag ,equal Tag, notEqual Tag, forward Tag, redirect Tag, greaterEqual Tag, greaterThan Tag, iterate Tag, lessEqual Tag, lessThan Tag, match Tag, notMatch Tag, present Tag, notPresent Tag

Suggested readings

1. James Goodwill,(2002). *Mastering Jakarta Struts*, Wiley Publishing, Inc.

Semester – III

17CAU311 DATA STRUCTURES PRACTICAL 4H – 2C

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

Course Objectives:
Enable the student

- To understand the fundamental concepts of data structures
- To Learn linear data structures – lists, stacks, and queues
- To apply Tree and Graph structures
- To understand sorting, searching and hashing algorithms
- To develop application using data structures
- Choose appropriate data structures and algorithms for problem solving.

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

1. Implement abstract data types for linear data structures.
2. Apply the different linear and non-linear data structures to problem solutions.
3. Analyze the applications of tree.
4. Implement graph theory over various data structures.
5. Critically analyze the various sorting algorithms.
6. Able to use linear and non-linear data structures like stacks, queues , linked list etc.

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. Write a program to scan a polynomial using linked list and add two polynomial.
11. Write a program to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
12. Write a program to display Fibonacci series (i)using recursion, (ii) using iteration
13. Write a program to calculate GCD of 2 number (i) with recursion (ii) without recursion
14. Write a program 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. Write a program to convert the Sparse Matrix into non-zero form and vice-versa.
16. Write a program to reverse the order of the elements in the stack using additional stack.
17. Write a program to reverse the order of the elements in the stack using additional Queue.
18. Write a program to implement Diagonal Matrix using one-dimensional array.
19. Write a program to implement Lower Triangular Matrix using one-dimensional array.
20. Write a program to implement Upper Triangular Matrix using one-dimensional array.
21. Write a program to implement Symmetric Matrix using one-dimensional array.
22. Write a program to create a Threaded Binary Tree as per inorder traversal, and implement operations like finding the suCSUessor / predecessor of an element, insert an element, inorder traversal.
23. Write a program to implement various operations on AVL Tree.

Semester – III

17CAU312 OPERATING SYSTEMS PRACTICAL 4H – 2C

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

Course Objectives:

Enable the student

- To Study the basic concepts and functions of operating systems.
- To understand the structure and functions of OS.
- To Learn about Processes, Threads and Scheduling algorithms.
- To Understand the principles of concurrency, Deadlocks and Memory Management
- To Learn about the Protection and Security Concepts.
- Understand basic resource management techniques.

Course Outcome (COs):

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

1. Design various Scheduling algorithms.
2. Apply the principles of concurrency.
3. Design deadlock, prevention and avoidance algorithms.
4. Compare and contrast various memory management schemes.
5. Apply the Security Concepts based on Authentication.
6. Appreciate the role of operating system as System software.

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 behavior of Linux kernel including CPU information and. memory information
3. Write a program to print file details including owner a access permissions, file access time, where file name is given as argument.
4. Write a program to copy files using system calls.
5. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.
6. Write program to implement FCFS scheduling algorithm.

Semester – III

17CAU312 OPERATING SYSTEMS PRACTICAL 4H – 2C

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.

Semester – III

17CAU313 COMPUTER NETWORKS PRACTICAL 4H – 2C

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

Course Objectives:

Enable the student

- To study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model.
- To read the fundamentals and basics concepts of Physical layer with real time examples
- To study data link layer concepts, design issues, and protocols.
- To learn the functions of network layer and the various routing protocols.
- To familiarize the functions and protocols of the Transport layer and Application layer.
- Channel error detection and correction, MAC protocols, Ethernet and WLAN.

Course Outcomes (COs):

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

- Describe the functions of each layer in OSI and TCP/IP model.
- Explain the functions of Application layer and Presentation layer paradigms and Protocols.
- Describe the Session layer design issues and Transport layer services.
- Classify the routing protocols and analyze how to assign the IP addresses for the given network.
- Describe the functions of data link layer and explain the protocols.
- 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.

Semester – III

17CAU314A ANDROID PROGRAMMING PRACTICAL 3H – 1C

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

Course Objectives:

Enable the student

- To compare the differences between Android and other mobile development environments.
- To learn the Object-oriented features of Kotlin and APIs for Android Development.
- To describe the working of Android applications, life cycle, manifest, and Intents
- To demonstrate the implementation of Form widgets for Android App development.
- To learn the SQLite database connectivity and database operations with android
- To design and develop useful Android applications with compelling user. interfaces .by using, extending, and creating your own layouts and Views and using Menus.

Course Outcome (Cos):

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

1. Analyze the Architecture and features of Android with another Mobile Operating System.
2. Evaluate the standard of Kotlin language for developing Android Applications
3. Apply knowledge for creating user Interface and develop activity for Android App.
4. Evaluate the user interface architecture of Android for developing Android Apps
5. Understand the implementation of SQLite database operations with Android.
6. Design and implement Database Application and Content providers.

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 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.

Semester – III

17CAU314B **STRUTS PRACTICAL** **3H – 1C**

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

Course Objectives:

Enable the students to

- Understand the Model, View, Controller (MVC) design pattern and how it is applied by Struts Framework
- Know the components of Struts Application and database connectivity
- Implement JSP functions using Struts
- Perform client and server side validation using Struts Validator Framework
- Develop web applications using Struts
- Develop EJB programs and get familiar with Struts framework

Course Outcome (COs):

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

1. Know about MVC and overview of JSP
2. Identify the components of a Struts Application and how to connect database in web based applications
3. Know about the struts Sub elements and Control Tags
4. Develop programs with Data Tags and Bean Tags
5. Develop programs with HTML Tags and Logic Tags and able to construct enterprise-level web based applications
6. Construct web based applications and Identify where data structures are appearing in them.

List of Programs:

1. Create a Simple Struts Human Resource (HR) Application

2. Create Struts Login Application Using Action form

3. Create a Strut Global Action Forwards and Action Mappings

4. Create and implement Multiple Struts Configuration File

5. Create Struts Application to implement struts Lookup Dispatch Action

6. Create Struts Application to mapping the struts Action Servlet

7. Create a Feedback form to implement struts HTML Tag

8. Create Struts Application to use of action forms for validating user input.

9. Create Struts Applications to implement Struts Logic Tag

10. Create a Struts LOGIC tag library provides tags that are useful in managing conditional generation of output text

Semester – IV

17CAU401 SOFTWARE ENGINEERING 4H – 4C

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

Course Objectives

- To introduce the fundamental concepts of software engineering.
- To Analyze, specify and document software requirements for a software system.
- To Design, select and apply the most appropriate software engineering process for a given project, plan for a software project, identify its scope and risks, and estimate its cost and time.
- Expose the criteria for test cases.
- Be familiar with test management and test automation techniques
- Implement a given software design using sound development practices.

Course Outcomes (COs)

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

1. Identify suitable life cycle models to be used and translate a requirement specification to a design using an appropriate software engineering methodology.
2. Apply systematic procedure for software design and deployment.
3. Analyze a problem and identify and define the computing requirements to the problem.
4. Formulate appropriate testing strategy for the given software system.
5. Develop software projects based on current technology, and test the software using testing tools.
6. An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives

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 EngineeringProcess- Requirement Analysis and Modeling Techniques- FlowOriented 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

Semester – IV

17CAU401 SOFTWARE ENGINEERING 4H – 4C

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

Websites

1. http://en.wikipedia.org/wiki/Software_engineering
2. http://www.onesmartclick.com/engineering/software-engineering.html
3. http://www.CSU.gatech.edu/classes/AY2000/cs3802_fall/

Semester – IV

17CAU402 DATA BASE MANAGEMENT SYSTEMS 4H – 4C

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

Course Objectives

Enable the student

- To describe a good introduction to the discipline of database management systems.
- To give a good formal foundation on the data models and E-R model.
- To demonstrate the principles database constraints behind systematic database design by covering normalization concept.
- To introduce the concepts of basic SQL as a universal Database language.
- To have an introductory knowledge about the PL/SQL concept
- To train the student to translate business requirements into relational database schemas and manipulate databases using the SQL Data Manipulation Language.

Course Outcomes (COs)

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

1. Demonstrate an understanding of the elementary features of RDBMS
2. Design conceptual models of a database using ER modeling for real life applications
3. Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database
4. Able to develop structured query language (SQL) queries to create, read, update, and delete relational database
5. Design efficient PL/SQL programs to access Oracle databases
6. Familiar with basic database storage structures and access techniques: file and page organizations, indexing methods including B tree, and hashing.

Unit-I

Introduction: Characteristics of database approach, data models, database system architecture and data independence. **Entity Relationship(ER) Modeling:** Entity types, relationships, constraints.

Unit-II

Relation data model: Relational model concepts, relational constraints, relational algebra.

Unit-III

Relation data model: SQL queries **Database design:** Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition.

Unit-IV

Database design: Normal forms (upto BCNF).**Transaction Processing :** ACID properties, concurrency control

Unit-V

File Structure and Indexing (8 Lectures) Operations on files, File of Unordered and ordered records, overview of File organizations, Indexing structures for files(Primary index, secondary index, clustering index), Multilevel indexing using B and B+ trees.

Suggested Readings

1. Elmasri, R., Navathe S.B., (2013). *Database Systems Models, Languages, Design and application Programming*, (6th ed.), Pearson Education.
2. Elmasri, R., Navathe, S.B.,(2010). *Fundamentals of Database Systems*, (6th ed.), Pearson Education.
3. Ramakrishanan,R., Gehrke,J., (2002). *Database Management Systems* (3rd ed.), McGraw-Hill.
4. Silberschatz,A., Korth, H.F., Sudarshan,S.,(2010). *Database System Concepts* (6th ed.), McGraw Hill.

Websites

1. http://en.wikipedia.org/wiki/RDBMS
2. http://aspalliance.com/1211_Relational_Database_Management_Systems__Concepts_and_Terminologies
3. www.compinfo-center.com/apps/rdbms.html

Semester – IV

17CAU403 **WEB PROGRAMMING** **4H – 4C**

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

Course Objectives

Enable the student

- To understand the concepts and architecture of the World Wide Web.
- To understand and practice markup languages
- Understand the technologies used in Web Programming.
- Know the importance of object-oriented aspects of Scripting.
- To understand and practice embedded dynamic scripting on client-side Internet Programming
- To develop dynamic web applications, create and consume web services

Course Outcomes (COs)

Upon Completion of the course, the students will be able to

1. Design web pages.
2. Use technologies of Web Programming.
3. Apply object-oriented aspects to Scripting.
4. Create a basic website using HTML and Cascading Style Sheets.
5. Design and implement dynamic web page with validation using JavaScript objects and by applying different event handling mechanisms.
6. Use scripting languages and web services to transfer data and add interactive components to web pages.

Unit-I

Creating an HTML Document – creating list – creating links between document – linking to resources on the internet – working with hypertext attributes – working with fonts and text styles – tables – creating frameset – working with forms – working with cascading style sheets.

Unit-II

JavaScript: Introduction to javascript – Programming fundamentals – Functions and objects – Navigator object model

Unit-III

JavaScript: Form and form elements – Scripting frames and multiple windows – Event object – Functions and custom objects.

Unit-IV

ASP: Client side scripting vs. Server side scripting- Variables & Constants- Procedures – Forms – Cookies – Application - #include – Global.asa - Functions-ASP object model: Response- Request- Application- Session – Server – Error – Array

Unit-V

ASP: Collections & Control Structure-File system object: File System – Text Stream- Drive – File – Folder – Directory – ADO - sql & Databases for data driven applications- ASP Components: Ad Rotator – Browser Cap. – Content Linking – Content Rotator .

Suggested Readings

1. Patrick Carey, (2005). *New Perspectives on HTML and XHTML*, (1st ed.), Thomson Course Technology Publishing. **(Unit- I).**
2. Rohit Khurana's, (2002). *Javascript Professional edition*, (2nd ed.), A.P.H. Publishing company, NewDelhi**.(Unit -II)**
3. Danny Goodman, (2000). *Javascript Bible*, (3rd ed.), IDG Books India Pvt Ltd. **(Unit- III).**
4. Russell Jones, A. (2000). *Mastering ActiveServerPages 3***, (**1st ed.), BPB Publishing, New Delhi.(**Unit- IV & Unit -V).**
5. David Flanagan, (2006). *JavaScript: The Definitive Guide*, O'Reilly,
6. Nicholas C. Zakas, Inc Ebrary and Ebrary,(2005). *Professional JavaScript for Web Developers*, New Delhi , John Wiley & Sons Inc.

Web Sites:

1. www.w3schools.com/
2. www.**javascript**kit.com
3. www.**asp**free.com
4. www.aspnettutorials.com

Semester – IV

17CAU404A R PROGRAMMING 3H – 3C

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

Course Objectives :

Enable the student

- To acquire the computing tasks such as using conditional processing statements, loops, and writing one's own functions.
- Performing advanced graphing of data and statistical modeling of data.
- Use statistical distribution functions in R
- Read Structured Data into R from various sources
- Understand split-apply-combine (group-wise operations) in R
- Perform basic statistical modeling of data

Course Outcome(COs):

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

1. Learn how to install and configure software necessary for a statistical programming environment.
2. Discuss generic programming language concepts as they are implemented in a high-level statistical language.
3. The course covers practical issues in statistical computing which includes programming in R, reading data into R, accessing R packages, writing R functions, debugging, and organizing and commenting R code.
4. Import external data into R for data processing and statistical analysis
5. Learn the main R data structures – vector and data frame
6. Learn the file processing in R

Unit-I

History and Overview of R : The S Philosophy - Back to R -Basic Features of R - Free Software -Design of the R System - Limitations of R- R Resources .Getting Started with R :Installation - Getting started with the R interface -.R Nuts and Bolts :Entering Input - Evaluation -R Objects - Numbers - Attributes - Creating Vectors - Mixing Objects - Explicit Coercion - Matrices -Lists -Factors - Missing Values - Data Frames - Names .

Semester – IV

17CAU404A R PROGRAMMING 3H – 3C

Unit-II

Getting Data In and Out of R :Reading and Writing Data - Reading Data Files with read.table() - Reading in Larger Datasets with read.table - Calculating Memory Requirements for R Objects . Using the readr Package .Using Textual and Binary Formats for Storing Data :Using dput() and dump() – Binary Formats - Interfaces to the Outside World : File Connections - Reading Lines of a Text File - Reading From a URL Connection - Subsetting R Objects :Subsetting a Vector - Subsetting a Matrix - Subsetting Lists - Subsetting Nested Elements of a List - Extracting Multiple Elements of a List - Partial Matching -Removing NA Values .

Unit-III

Vectorized Operations :Vectorized Matrix Operations .Dates and Times :Dates in R - Times in R - Operations on Dates and Times .Managing Data Frames with the dplyr package :Data Frames -The dplyr Package - dplyr Grammar - Installing the dplyr package - select() - filter() -arrange() - rename() - mutate() - group_by()-%>%.Control Structures :if-else - for Loops - Nested for loops - while Loops - repeat Loops - next, break .

Unit-IV

Functions: Functions in R - Your First Function - Argument Matching - Lazy Evaluation – The Argument - Arguments Coming After the Argument .Scoping Rules of R : A Diversion on Binding Values to Symbol - Scoping Rules - Lexical Scoping: Why Does It Matter? -Lexical vs. Dynamic Scoping -- Application: Optimization - Plotting the Likelihood. Coding Standards for R .Loop Functions : Looping on the Command Line - lapply() - sapply() - split() - Splitting a Data Frame - tapply - apply() - Col/Row Sums and Means -Other Ways to Apply - mapply()-Vectorizing a Function .

Unit-V

Debugging -:Something’s Wrong! - Figuring Out What’s Wrong - Debugging Tools in R . Using traceback() - Using debug() - Using recover().Profiling R Code: Using system.time() . Timing Longer Expressions - The R Profiler - Using summaryRprof().Simulation :Generating Random Numbers - Setting the random number seed -Simulating a Linear Model - Random Sampling .

Suggested Readings

1. Daniel Navarro, (2013). *Learning Statistics with R.* University of Adelaide Publications.
2. Hadley Wickham, (2014). *Advanced R Programming*, (1st ed.)

3. Jeffrey Stanton, (2013). *Introduction to Data Science, with Introduction to R*, Version 3 ,
4. Roger.D.Peng, (2015). *R Programming for Data Science*

Semester – IV

17CAU404B **SOFTWARE TESTING** **3H – 3C**

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

Course Objectives:

To help students:

- to understand the strengths and weaknesses of a variety of software testing techniques.
- to get familiar with the techniques, and tools in the area of software testing and its practice in the industry
- to become a good software tester
- to gain confidence in and providing information about the level of quality of the software
- to find defects which may get created by the programmer while developing the software.
- Know to develop software

Course Outcome:

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. Test the various types in the software application

Unit- I

Assessing Testing Capabilities and Competencies

Who is associated with testing? – The multiple Roles of Testing – Defect – The business perspective for testing – test process and testers ; Building a software testing strategy: Computer System Strategic risks – Economics of Testing – Common Computer Problems – Economics of System Development Life Cycle (SDLC) Testing – Testing-an Organizational Issue – Establishing a testing policy – Structured approach to testing – Test Strategy; Software Testing Methodology: Reduce the cost of testing – Functional and Structural testing – Workbench concept – Eight considerations in developing testing methodologies

Semester – IV

17CAU404B **SOFTWARE TESTING** **3H – 3C**

Unit -II

Software 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; 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.

Unit- III

Software Testing Techniques

Determining your software testing techniques: Testing Techniques/Tools selection process – Selecting Techniques/tools – Structural system testing techniques- Functional system testing techniques – Unit testing technique – Functional testing and analysis – Functional testing – Test factor/Test technique matrix

Testing process

The Cost of Computer Testing – Life Cycle Testing concept – Verification and validation in the software. Assess Project Management Development Estimate and Status - Develop Test Plan - Requirements Phase Testing -Design Phase Testing -Program Phase Testing

Unit -IV

Testing process

Execute Test and Record results- Acceptance Test- Report Test Results- Testing Software Installation- Test Software Changes - Evaluate Test Effectiveness.

Unit- V

Testing Specialized Systems and Applications

Testing client/server systems - Testing web-based systems - Testing security - Building tests documentation.

Suggested Readings

1. Ron Patton, (2004). *Software Testing*, (2nd ed.), New Delhi, Pearson Education. (2nd unit)

2. William E.Perry, (2001). *Effective methods for Software Testing*, (2nd ed.), New Delhi: John Wiley & Sons, Inc., (1, 3, 4 & 5 units)

Semester – IV

17CAU411 SOFTWARE ENGINEERING PRACTICAL 4H – 2C

Course Objectives

- To introduce the fundamental concepts of software engineering.
- To Analyze, specify and document software requirements for a software system.
- To Design, select and apply the most appropriate software engineering process for a given project, plan for a software project, identify its scope and risks, and estimate its cost and time.
- Expose the criteria for test cases.
- Be familiar with test management and test automation techniques
- Implement a given software design using sound development practices.

Course Outcomes (COs)

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

1. Identify suitable life cycle models to be used and translate a requirement specification to a design using an appropriate software engineering methodology.
2. Apply systematic procedure for software design and deployment.
3. Analyze a problem and identify and define the computing requirements to the problem.
4. Formulate appropriate testing strategy for the given software system.
5. Develop software projects based on current technology, and test the software using testing tools.
6. An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives

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

Semester – IV

17CAU412 DATA BASE MANAGEMENT SYSTEMS PRACTICAL 4H – 2C

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

Course Objectives

Enable the student

- To describe a good introduction to the discipline of database management systems.
- To give a good formal foundation on the data models and E-R model.
- To demonstrate the principles database constraints behind systematic database design by covering normalization concept.
- To introduce the concepts of basic SQL as a universal Database language.
- To have an introductory knowledge about the PL/SQL concept
- To train the student to translate business requirements into relational database schemas and manipulate databases using the SQL Data Manipulation Language.

Course Outcomes (COs)

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

1. Demonstrate an understanding of the elementary features of RDBMS
2. Design conceptual models of a database using ER modeling for real life applications
3. Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database
4. Able to develop structured query language (SQL) queries to create, read, update, and delete relational database
5. Design efficient PL/SQL programs to access Oracle databases
6. Familiar with basic database storage structures and access techniques: file and page organizations, indexing methods including B tree, and hashing.

Semester – IV

17CAU412 DATA BASE MANAGEMENT SYSTEMS PRACTICAL 4H – 2C

List of Programs:

1.Create and use the following database schema to answer the given queries.

EMPLOYEE Schema

Field	**Type**	**NULL**	**KEY**	**DEFAULT**
Eno	Char(3)	NO	PRI	NIL
Ename	Varchar(50)	NO		NIL
Job_type	Varchar(50)	NO		NIL
Manager	Char(3)	Yes	FK	NIL
Hire_date	Date	NO		NIL
Dno	Integer	YES	FK	NIL
Commission	Decimal(10,2)	YES		NIL
Salary	Decimal(7,2)	NO		NIL

DEPARTMENT Schema

Field	**Type**	**NULL**	**KEY**	**DEFAULT**
Dno	Integer	No	PRI	NULL
Dname	Varchar(50)	Yes		NULL
Location	Varchar(50)	Yes		New Delhi

Query List

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than $2850
6. Query to display Employee Name and Department Number for the Employee No= 7900.

7. Query to display Employee Name and Salary for all employees whose salary is not in the range of $1500 and $2850.

8. Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.

9. Query to display Name and Hire Date of every Employee who was hired in 1981.

10. Query to display Name and Job of all employees who don‟t have a current Manager.

11. Query to display the Name, Salary and Commission for all the employees who earn commission.

12. Sort the data in descending order of Salary and Commission.

13. Query to display Name of all the employees where the third letter of their name is ‗A‟.

14. Query to display Name of all employees either have two ‗R‟s or have two ‗A‟s in their name and are either in Dept No = 30 or their Manger‟s Employee No = 7788.

15. Query to display Name, Salary and Commission for all employees whose Commission Amount is 14 greater than their Salary increased by 5%.

16. Query to display the Current Date.

17. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.

18. Query to display Name and calculate the number of months between today and the date each employee was hired.

19. Query to display the following for each employee <E-Name> earns < Salary> monthly but wants < 3 * Current Salary >. Label the Column as Dream Salary.

20. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with ‗J‟, ‗A‟ and ‗M‟.

21. Query to display Name, Hire Date and Day of the week on which the employee started.

22. Query to display Name, Department Name and Department No for all the employees.

23. Query to display Unique Listing of all Jobs that are in Department # 30.

24. Query to display Name, Dept Name of all employees who have an _A‘ in their name.

25. Query to display Name, Job, Department No. And Department Name for all the employees working at the Dallas location.

26. Query to display Name and Employee no. Along with their Manger‘s Name and the Manager‘s employee no; along with the Employees‘ Name who do not have a Manager.

27. Query to display Name, Dept No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns a commission.

28. Query to display Name and Salaries represented by asterisks, where each asterisk (*) signifies $100.

29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees

30. Query to display the number of employees performing the same Job type functions.

31. Query to display the no. of managers without listing their names.

32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.

33. Query to display Name and Hire Date for all employees in the same dept. as Blake.

34. Query to display the Employee No. And Name for all employees who earn more than the average salary.

35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a _T‘.

36. Query to display the names and salaries of all employees who report to King.

37. Query to display the department no, name and job for all employees in the Sales department.

Semester – IV

17CAU413 WEB PROGRAMMING PRACTICAL 4H – 2C

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

Course Objectives

Enable the student

- To understand the concepts and architecture of the World Wide Web.
- To understand and practice markup languages
- Understand the technologies used in Web Programming.
- Know the importance of object-oriented aspects of Scripting.
- To understand and practice embedded dynamic scripting on client-side Internet Programming
- To develop dynamic web applications, create and consume web services

Course Outcomes (COs)

Upon Completion of the course, the students will be able to

1. Design web pages.
2. Use technologies of Web Programming.
3. Apply object-oriented aspects to Scripting.
4. Create a basic website using HTML and Cascading Style Sheets.
5. Design and implement dynamic web page with validation using JavaScript objects and by applying different event handling mechanisms.
6. Use scripting languages and web services to transfer data and add interactive components to web pages.

List of Programs:

1. Develop a website for Karpagam University using HTML.
2. Write Online Quiz program (Include Style Sheets)
3. Create a simple animation using DHTML.
4. Write a program to apply Mask into an Image Using Filters in DHTML.
5. Generate web page that represents clock-every 60 see the page updated with server current time Using JavaScript.
6. Design a form and validate it using JavaScript.
7. Show the demo of AD Rotator Component.
8. Write Database Access program using ASP.
9. Program to retrieve Cookies information using ASP

Semester – IV

17CAU413 WEB PROGRAMMING PRACTICAL 4H – 2C

10. Program to count web page hits using ASP
11. Program to create Date & Time, String Manipulation using ASP
12. Write a program to find the visitor’s Browser Type, IP Address and More Information

Semester – IV

17CAU414A R PROGRAMMING (PRACTICAL) 3H – 1C

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

Course Objectives :

Enable the student

- To acquire the computing tasks such as using conditional processing statements, loops, and writing one's own functions.
- Performing advanced graphing of data and statistical modeling of data.
- Use statistical distribution functions in R
- Read Structured Data into R from various sources
- Understand split-apply-combine (group-wise operations) in R
- Perform basic statistical modeling of data

Course Outcome(COs):

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

1. Learn how to install and configure software necessary for a statistical programming environment.
2. Discuss generic programming language concepts as they are implemented in a high-level statistical language.
3. The course covers practical issues in statistical computing which includes programming in R, reading data into R, accessing R packages, writing R functions, debugging, and organizing and commenting R code.
4. Import external data into R for data processing and statistical analysis
5. Learn the main R data structures – vector and data frame
6. Learn the file processing in R

List of Programs:

1. Write a program to demonstrate functions and operators
2. **Vectors:** Grouping values into vectors, then doing arithmetic and graphs with them
3. **Matrices:** Creating and graphing two-dimensional data sets
4. **Summary Statistics:** Calculating and plotting some basic statistics: mean, median, and standard deviation

5. **Factors:** Creating and plotting categorized data
6. **Data Frames:** Organizing values into data frames, loading frames from files and merging them
7. Write a program to design R as a calculator
8. Write a program to demonstrate Probability distributions
9. Write a program to demonstrate Importing and exporting data
10. Write a program to Establish a Regression

Semester – IV

17CAU414B **Software Testing PRACTICAL** **3H – 1C**

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

Course objective

Enable the student

- Basic software debugging methods.
- White box testing methods and techniques.
- Black Box testing methods and techniques.
- Designing test plans.
- Different testing tools (familiar with open source tools)
- Quality Assurance models.

Course outcome

Upon the completion of the course, student able to

1. Apply modern software testing processes in relation to software development and project management.
2. Create test strategies and plans, design test cases, prioritize and execute them.
3. Manage incidents and risks within a project.
4. Contribute to efficient delivery of software solutions and implement improvements in the software development processes.
5. To gain expertise in designing
6. implement and development of computer based systems and IT processes.

List of Programs:

1. Create a VB form with the following fields and create the database also for them. Insert 3 records. Using Win Runner tool record the above 3 transaction and test them and produce the Report. (Blackbox Testing).

2. Create a VB form and then add login dialog form. Using Win Runner tool check the Username and Password and produce the Report. (Security testing).

3. Create a VB form with the following fields and check the calculation is correct or not by using the test toll Win Runner. (Functional Testing) Fields – Name, Designation, Department, Basic, HRA, DA, PF and netsal.

4.using Win Runner test tool check the database values after changing. Using Flight database. (Regression testing).

5. Write a C program for Boundary Testing.

6. Write a C program for Loop Testing.

7. Write a C program for Integration Testing.

8. Write a C program for Interface Testing.

9. Write a C program for Unit testing.

Semester – V

17CAU501A **Oracle (SQL / PL-SQL)** **3H – 3C**

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

Course Objectives

Enable the student

- create and use table, column, primary key, foreign key,
- Use constraint, data type and view.
- Use SQL DDL commands to on a very basic level create and edit an Oracle database.
- Use SQL DML commands to select, update and delete data in an Oracle database.
- Use built-in functions and dynamic SQL in an Oracle 10g database.
- Use PL/SQL to build procedures, functions and triggers with variables, cursors, flow control and error management in Oracle 10g database

Course Outcomes:

Upon the completion of the course, student able to

1. Develop efficient PL/SQL programs to access Oracle databases
2. Use some of the Oracle supplied PL/SQL packages to generate screen and file outputs
3. Design modular applications using packages
4. Invoke native dynamic SQL to build runtime SQL statements
5. Manage data retrieval with cursors and cursor variables
6. Enhance performance using collection datatypes and bulk operations

Unit-I

Introduction to Oracle as RDBMS SQL Vs. SQL * Plus: SQL Commands and Data types, Operators and Expressions, Introduction to SQL * Plus.

Unit-II

Managing Tables and Data: Creating and Altering Tables (Including constraints) ,Data Manipulation Command like Insert, update, delete, SELECT statement with WHERE, GROUP BY and HAVING, ORDER BY, DISTINCT, Special operator like IN, ANY, ALL BETWEEN, EXISTS, LIKE, Join, Built in functions

Unit-III

Other Database Objects - View, Synonyms, Index

Unit-IV

Transaction Control Statements - Commit, Rollback, Savepoint

Unit-V

Introduction to PL/SQL SQL v/s PL/SQL, PL/SQL Block Structure, Language construct of PL/SQL (Variables, Basic and Composite Data type, Conditions looping etc.) TYPE and % ROWTYPE , Using Cursor (Implicit, Explicit)

Suggested Readings

1. Ivan Bayross, (2010). SQL, *PL/SQL the Programming Language of Oracle Paperback*, BPB Publicatins.

2. John Watson, Roopesh Ramklass,(2008). *OCA Oracle Database11g SQL Fundamentals I Exam Guide*, Oracle Press.

3. Michael Mclaughlin,(2008). *Oracle Database 11g PL/SQL Programming*, Oracle Press.

4. Rajeeb C. Chatterjee,(2012). *Learning Oracle SQL and PL/SQL: A simplified Guide*, PHI.

5. Ron Hardman, Michael Mclaughlin, (2005). *Expert Oracle PL/SQL*, Oracle Press.

6. Steven Feuerstein, Bill Pribyl, (2014). *Oracle PL/SQL Programming*, (6th ed.), O'Reilly Media.

Semester – V

17CAU501B PROGRAMMING IN PYTHON 3H – 3C

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

Course Objectives

- To Learn Syntax and Semantics and create Functions in Python.
- To Understand the basic logic statements in Python
- To Handle Strings in Python.
- To Understand Lists, Dictionaries in Python.
- To Build GUI applications
- To Use Python interactively

Course Outcomes (COs)

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

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

Unit-I

Planning the Computer Program: Concept of problem solving-Problem definition-Program design-Debugging-Types of errors in programming-Documentation.

Unit-II

Techniques of Problem Solving: Flowcharting-decision table-algorithms-Structured programming concepts-Programming methodologies: top-down and bottom-up Programming.

Unit-III

Overview of Programming: Structure of a Python Program-Elements of Python.

Unit-IV

Introduction to Python: Python Interpreter-Using Python as calculator-Python shell-Indentation. Atoms-Identifiers and keywords-Literals-Strings-Operators(Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator).

Semester – V

17CAU501B PROGRAMMING IN PYTHON 3H – 3C

Unit-V

Creating Python Programs: Input and Output Statements-Control statements(Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.). Defining Functions-Default arguments.

Suggested Readings

1. Allen Downey, Jeffrey Elkner, Chris Meyers, (2012). How to think like a computer scientist : learning with Python , Freely available online.
2. Budd,T.,(2011). *Exploring Python*, (1st ed.) TMH

Websites

1. http://docs.python.org/3/tutorial/index.html.
2. http://interactivepython.org/courselib/static/pythonds.
3. http://www.ibiblio.org/g2swap/byteofpython/read/.

Semester – V

17CAU502A **INFORMATION SECURITY** **4H – 4C**

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

Course Objectives

- To provides an overview of Information Security and Assurance.
- To provide an exposure to the spectrum of security activities methods methodologies and procedures with emphasis on practical aspects of Information Security.
- Gain knowledge about securing both clean and corrupted systems, protect personal data, and secure computer networks.
- Understand key terms and concepts in cyber law, intellectual property and cyber crimes, trademarks and domain theft.
- To examine secure software development practices.
- To incorporate approaches for incident analysis and response.
- To incorporate approaches for risk management and best practices.

Course Outcomes(COs)

A student who successfully completes this course should at a minimum be able to:

1. State the basic concepts in information security
2. Explain concepts related to applied cryptography including the four techniques for crypto-analysis symmetric and asymmetric cryptography, digital signature, message authentication code, hash functions and modes of encryption operations.
3. Explain common vulnerabilities in computer programs including buffer overflow Vulnerabilities time-of-check to time-of-use flaws incomplete mediation.
4. The learner will gain an understanding of cryptography, how it has evolved, and some key encryption techniques used today.
5. The learner will develop an understanding of security policies (such as confidentiality, integrity, and availability), as well as protocols to implement such policies.
6. The learner will gain familiarity with prevalent network and distributed system attacks, defenses against them, and forensics to investigate the aftermath.

Unit -I

Introduction: Security, Attacks, Computer Criminals, Security Services, Security Mechanisms.

Cryptography : Substitution ciphers, Transpositions Cipher, Confusion, diffusion.

Unit – II

Symmetric, Asymmetric Encryption. DES Modes of DES, Uses of Encryption, Hash function, key exchange, Digital Signatures, Digital Certificates

Unit – III

Program Security: Secure programs, Non malicious Program errors, Malicious codes virus, Trap doors, Salami attacks, Covert channels, Control against program.
Threats: Protection in OS: Memory and Address Protection, Access control, File Protection, User Authentication**.**

Unit – IV

Database Security: Requirements, Reliability, Integrity, Sensitive data, Inference, Multilevel Security**.**
Security in Networks : Threats in Networks, Security Controls, firewalls, Intrusion detection systems, Secure e-mails

Unit -V

Administrating Security: Security Planning, Risk Analysis, Organizational Security Policy, Physical Security. Ethical issues in Security: Protecting Programs and data. Information and law.

Suggested Readings

1. Pfleeger, C. P. , Pfleeger, S. L.,(2006). Security in Computing, Prentice Hall of India.

2. Stallings,W., (2010).Network Security Essentials: Applications and Standards, (4th ed.).

Semester – V

6CAU502B **DATA MINING** **4H – 4C**

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

Course Objectives

- To Understand Data Mining fundamentals and Characterize the kinds of patterns that can be discovered by association rule mining
- To Compare and evaluate different data mining techniques like classification, prediction.
- To Cluster the high dimensional data for better organization of the data
- To describe complex data types with respect to spatial and web mining
- To Design data warehouse with dimensional modelling and apply OLAP operations.
- To program using available data mining tools and general-purpose languages.

Course Outcomes(COs)

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

1. Extract knowledge using data mining techniques and Implement Preprocess the data for mining applications and apply the association rules for mining the data
2. Design and deploy appropriate classification techniques and decision trees.
3. Understand the concept of clustering and its real time applications
4. Explore recent trends in data mining such as web mining, spatial-temporal mining
5. Able to know the basic concepts of data warehouse and OLAP operations
6. Organize and Prepare the data needed for data mining using pre preprocessing techniques

Unit- I

Introduction : Fundamentals of data mining – Data Mining Functionalities – Classification of Data Mining systems – Major issues in Data Mining.
Data Warehouse and OLAP Technology: An Overview – Data Warehouse – Multidimensional Data Model – Data Warehouse Architecture

Unit-II

Data Preprocessing: Needs Preprocessing the Data – Data Cleaning – Data Integration and Transformation – Data Reduction – Discretization and Concept Hierarchy Generation – Online Data Storage.

Preparing Data for Mining: Variable Measures.

Unit-III

Mining Frequent Patterns, Associations and Correlations: Basic Concepts – Efficient and Scalable Frequent item set Mining Methods – From Association Mining to Correlation Analysis.

Unit-IV

Predictive and descriptive data mining techniques, supervised and unsupervised learning techniques, process of knowledge discovery in databases, pre-processing methods

Unit-V

Data Mining Techniques: Association Rule Mining, classification and regression techniques, clustering, Scalability and data management issues in data mining algorithms, measures of interestingness

Suggested Readings

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

Semester – V

17CAU503A **MICROPROCESSOR** **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:

Enable the student

- To become familiar with the architecture and the instruction set of an Intel microprocessor
- To illustrate the architecture of 8085 and 8086 microprocessors.
- To introduce the programming and interfacing techniques of 8086 microprocessor.
- To analyse the basic concepts and programming of 8051 microcontroller
- To understand the interfacing circuits for various applications of 8051 microcontroller.
- To introduce the architecture of advanced microprocessors and microcontrollers.

Course Outcomes:

Upon the completion of the course, the student will able to:

1. Design and implement programs on 8086, ARM, PIC. CO2 Design I/O circuits.
2. The program prepares students to successfully compete for employment in Electronics, Manufacturing and Embedded fields.
3. Design Memory Interfacing circuits.
4. Design and implement 8051 microcontroller based systems.
5. Describe the architecture and instruction set of ARM microcontroller
6. Assembly language programming will be studied as well as the design of various types of digital and analog interfaces

Unit – I

Introduction to Microprocessor: Introduction to 8085 – Pin Diagram – Microprocessor architecture and its operations– Demultiplexing the Bus – Generation of Control Signals – Fetching, Decoding and Execution of Instruction – Instruction Timing and Status Flag.

Unit – II

Addressing Modes: Instruction Set – Addressing Modes – Instruction Format – Simple Program – Memory and Machine Cycle – Memory Units Machine Cycle.

Unit – III

Interfacing Concepts: Peripheral I/O Instructions – Device Selection and Data Transfer – Types of Data Transfer – Input Interfacing – Input Interfacing using Decoders – Output Interfacing – LED and 7 Segment Display – Interfacing Memory.

Unit – IV

Peripheral Devices: Introduction to Programmable Peripheral Interface 8255 – Pin Diagram – Architecture – Modes of Operation: I/O and BSR – Architecture and Operation of 8251 (USART)

Architecture and Operation of Programmable Interrupt Controller (8259) – Architecture of 254(8253) Programmable Interval Timer/ Counter – DMA Controller(8259).

Unit – V - Applications

Delay Program – Traffic Light Control System – Water Level Controller – Stepper Motor Control – Interfacing DAC – Interfacing ADC – Temperature Measurement.

Suggested Readings:

1. Barry B. Brey, (2009). *The Intel Microprocessors : Architecture, Programming and Interfacing*. (6th ed.)Pearson Education.
2. Gupta,M.K.,(2006). *Microprocessor Microcomputer, Microcontroller and Interfacing*, (1st ed.), New Delhi , Paragon International Publisher.
3. Ramesh S.Gaonkar, (2000). *Microprocessor Architecture, Programming and Application with 8085*, (4th ed.), New Delhi, Penram International Publishing.
4. Walter A Triebel, Avtar Singh, (2005). *The 8088 and 8086 Microprocessors Programming, Interfacing, Software, Hardware, and Applications*. (4th ed.)PHI,

17CAU503B PC HARDWARE AND TROUBLE SHOOTING 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:

Enable the student

- to assemble/setup and upgrade personal computer systems
- perform installation, configuration,
- upgrading of hardware and software,
- install/connect associated peripherals
- diagnose in hardware and software and other peripheral equipment.
- Troubleshoot the problem

Course Outcomes:

Upon the completion of the course, the student will be able to

1. assemble/setup and upgrade personal computer systems
2. perform installation, configuration,
3. upgrading of hardware and software,
4. install/connect associated peripherals
5. diagnose in hardware and software and other peripheral equipment.
6. Troubleshoot the problem

Unit-I

Micro Computer System: Introduction to Micro Computer System – Computer Organization – Number Systems and Codes Memory – Arithmetic and Logic Unit – Control Unit.

Unit-II

Peripheral Devices: Introduction to Peripheral Devices – Keyboard – CRT Display monitor – Printer – Magnetic Storage Devices – Floppy Disk Drive – Hard Disk Drive – Peripherals Interfaces and Controller – Keyboard Interface

17CAU503B PC HARDWARE AND TROUBLE SHOOTING 4H – 4C

Unit-III

Display Adapter: CRT Display — CRT Controller –Auxiliary Subsystems – Data Communication fundamentals – Serial Port in PC – Real time clock (RTC) – Magnetic Tape Subsystems – LAN – Memory Expansion Options

Unit-IV

Installation and Preventive Maintenance: Pre Installation Planning – Installation Practice – Routine Checks – Special Configurations – Memory Up Gradation

Unit-V

Trouble shooting: Troubleshooting – Computer faults – Nature of faults – Types of Faults Diagnostic Programs and Tools — Faults in Elimination Process – Systematic Troubleshooting – POST (Power on Self Test)

Suggested Readings

1. Govindarajalu,B.,(2011). *IBM PC and Clones*,(2nd ed.)Tata McGraw Hill Publishing Company. [UNIT I & II]
2. Michael Meyers, (2003). *Introduction to PC Hardware and Troubleshooting, The Mike Meyers' Computer Skills*, (1st ed.)McGraw Hill, [UNIT III - V]
3. Sanjay K. Bose, (1999). *Hardware and Software of Personal Computers*, New Age International Publishers.

Semester – V

17CAU504A DIGITAL IMAGE PROCESSING 4H – 4C

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

Course Objectives

- 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 apply knowledge of mathematics, science, and engineering
- To design and conduct experiments, as well as to analyze and interpret data

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 deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Adaptive filters, Linear, Position invariant degradations, Estimation of Degradation functions, Restoration from projections, 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, Sub-image size selection, blocking artifacts, DCT implementation using FFT, Run length coding.

Unit – IV

FAX compression (CSUITT Group-3 and Group-4), Symbol-based coding, JBIG-2, Bit-plane encoding, Bit-allocation, Zonal Coding, Threshold Coding, JPEG, Lossless predictive coding, Lossy predictive coding, Motion Compensation
Wavelet based Image Compression: Expansion of functions, Multi-resolution analysis, Scaling functions, MRA refinement equation, 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, Hit-or-Miss Transform, Boundary Detection, Hole filling, Connected components, convex hull, thinning, thickening, skeletons, pruning, Geodesic Dilation, Erosion, Reconstruction by dilation and erosion. Image Segmentation: Boundary detection based techniques, Point, line detection, Edge detection, Edge linking, local processing, regional processing, Hough transform, Thresholding, Iterative thresholding, Otsu's method, Moving averages, Multivariable thresholding, Region-based segmentation, Watershed algorithm, Use of motion in segmentation

Suggested Readings

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

Semester – V

17CAU504B OPEN SOURCE TECHNOLOGIES 4H – 4C

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

Course Objectives:

Enable the student

- To introduce open source technology for development of web applications.
- For Study the problems with traditional commercial software.
- To understand open source scripting language for programming in web environment i.e. PHP.
- To study the open source management system and connection with database.
- To learn open source web server, software tools.
- To learn the open source ideals in order to apply those principles

Course Outcomes (COs)

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

1. Leaned the need of open source technology, open source development model, application of open sources, aspects of open source movement
2. The students will be aware about the problems with traditional commercial software.
3. The student will be familiar with basis syntax of PHP, common PHP scripts elements.
4. The student will be familiar with creating of the server side scripting using PHP, implement PHP database connectivity, perform operation on database and open source database management system.
5. The students will be familiar with Working of different web Servers.
6. The students will be aware about the software tool and process like Eclipse IDE, Selenium ID

Unit-I

Why open source, what is Open source, open source principles, open standards requirements for software, open source successes, free software, some example of free software, free software license provider, free software Vs Open source software, Public Domain , FOSS DOES not Mean any cost, proprietary Vs Open Source Licensing Model. Principles and Open Source Methodology: History, open source initiatives, open standards principles-methodologies, philosophy, software freedom, open source software, development, Licenses, copyright.

Unit-II

Open source projects: Starting and maintaining an open source project, open source hardware- open source design-open source teaching (OST).Open Source Ethics: Open Source Vs Closed Source-Open source Government-The ethics of open source-social and financial impacts of open source technology-shared software, shared source.

Unit-III

Apache Web Server: Introduction-Starting, Stopping, and Restarting Apache-Configuration-Securing Apache Create the Web Site-Apache Log Files

Unit-IV

MySQL: Introduction-Tutorial-Database Independent Interface-Table Joins-Loading and Dumping a Database

Unit-V

Perl: Introduction-Perl Documentation-Perl Syntax Rules-A Quick Introduction To Object-Oriented Programming-What We Didn't Talk About

Suggested Readings

1. Andrew M. St. Laurent, (2004). *Understanding Open Source and Free Software Licensing*, O'Reilly Media.
2. Dan Woods, Gautam Guliani, (2005). Open Source for the Enterprise, O'Reilly Media.
3. Fadi P. Deek and James A. M. McHugh,(2007). *Open Source Technology and Policy*, Cambridge University Press.
4. James Lee, Brent Ware,2002. *Open Source Web Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP*, , Publisher: Addison Wesley Date [UNIT III – V]
5. Kailashvadera, Vhavyesh Gandhi, (2009). *Open source Technologies, Lakshmi Publications*,(1st ed.). [UNIT I – II]
6. Nick Wells,(2012). *The Complete Guide to Linux System Administration*, Delmar Cengage Learning.

Semester – V

17CAU511A Oracle (SQL/PL-SQL) (PRACTICAL) 3H – 1C

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

Course Objectives

Enable the student

- create and use table, column, primary key, foreign key,
- Use constraint, data type and view.
- Use SQL DDL commands to on a very basic level create and edit an Oracle database.
- Use SQL DML commands to select, update and delete data in an Oracle database.
- Use built-in functions and dynamic SQL in an Oracle 10g database.
- Use PL/SQL to build procedures, functions and triggers with variables, cursors, flow control and error management in Oracle 10g database

Course Outcomes:

Upon the completion of the course, student able to

1. Develop efficient PL/SQL programs to access Oracle databases
2. Use some of the Oracle supplied PL/SQL packages to generate screen and file outputs
3. Design modular applications using packages
4. Invoke native dynamic SQL to build runtime SQL statements
5. Manage data retrieval with cursors and cursor variables
6. Enhance performance using collection datatypes and bulk operations

[SQL COMMANDS]

1) SQL* formatting commands

2) To create a table, alter and drop table.

3) To perform select, update, insert and delete operation in a table.

4) To make use of different clauses viz where, group by, having, order by, union and intersection,

5) To study different constraints.

[SQL FUNCTION]

6) To use oracle function viz aggregate, numeric, conversion, string function.

7) To understand use and working with joins.

8) To make use of transaction control statement viz rollback, commit and save point.

9) To make views of a table.

10) To make indexes of a table.

[PL/SQL]

11) To understand working with PL/SQL

12) To implement Cursor on a table.

13) To implement trigger on a table

Semester – V

17CAU511B PROGRAMMING IN PYTHON (PRACTICAL) 3H – 1C

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

Course Objectives

Enable the student

- To Learn Syntax and Semantics and create Functions in Python.
- To Understand the basic logic statements in Python
- To Handle Strings in Python.
- To Understand Lists, Dictionaries in Python.
- To Build GUI applications
- To Use Python interactively

Course Outcomes (COs)

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

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

List of Programs:

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's choice.
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 aCSUording to the following criteria :
   Grade A: Percentage >=80
   Grade B: Percentage>=70 and <80
   Grade C: Percentage>=60 and <70
   Grade D: Percentage>=40 and <60
   Grade E: Percentage<40
3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by aCSUepting suitable input parameters from user.
4. Write a program in python to display the first n terms of Fibonacci series.
5. Write a program in python to find factorial of the given number.
6. Write a program in python to find sum of the following series for n terms: 1 – 2/2! + 3/3! - - - - - n/n!
7. Write a program in python to calculate the sum and product of two compatible matrices.

Semester – V

17CAU512A INFORMATION SECURITY PRACTICAL 4H – 2C

Instruction Hours / week: L: 0 T: 0 P: 2 **Marks:** Int : **40** Ext : **60** Total: **100**

Course Objectives

- To provides an overview of Information Security and Assurance.
- To provide an exposure to the spectrum of security activities methods methodologies and procedures with emphasis on practical aspects of Information Security.
- Gain knowledge about securing both clean and corrupted systems, protect personal data, and secure computer networks.
- Understand key terms and concepts in cyber law, intellectual property and cyber crimes, trademarks and domain theft.
- To examine secure software development practices.
- To incorporate approaches for incident analysis and response.
- To incorporate approaches for risk management and best practices.

Course Outcomes(COs)

A student who successfully completes this course should at a minimum be able to:

1. State the basic concepts in information security
2. Explain concepts related to applied cryptography including the four techniques for crypto-analysis symmetric and asymmetric cryptography, digital signature, message authentication code, hash functions and modes of encryption operations.
3. Explain common vulnerabilities in computer programs including buffer overflow Vulnerabilities time-of-check to time-of-use flaws incomplete mediation.
4. The learner will gain an understanding of cryptography, how it has evolved, and some key encryption techniques used today.
5. The learner will develop an understanding of security policies (such as confidentiality, integrity, and availability), as well as protocols to implement such policies.
6. The learner will gain familiarity with prevalent network and distributed system attacks, defenses against them, and forensics to investigate the aftermath.

List of Programs:

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois
2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the trength of passwords using these tools.
3. Perform encryption and decryption of Caesar cipher. Write a script for performing these operations.
4. Perform encryption and decryption of a Rail fence cipher. Write a script for

performing these operations.

5. Use nmap/zenmap to analyse a remote machine.
6. Use Burp proxy to capture and modify the message.
7. Demonstrate sending of a protected word document.
8. Demonstrate sending of a digitally signed document.
9. Demonstrate sending of a protected worksheet.
10. Demonstrate use of steganography tools.
11. Demonstrate use of gpg utility for signing and encrypting purposes.

Semester – V

17CAU512B **DATA MINING 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

Enable the student

- To Understand Data Mining fundamentals and Characterize the kinds of patterns that can be discovered by association rule mining
- To Compare and evaluate different data mining techniques like classification, prediction.
- To Cluster the high dimensional data for better organization of the data
- To describe complex data types with respect to spatial and web mining
- To Design data warehouse with dimensional modelling and apply OLAP operations.
- To program using available data mining tools and general-purpose languages.

Course Outcomes(COs)

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

1. Extract knowledge using data mining techniques and Implement Preprocess the data for mining applications and apply the association rules for mining the data
2. Design and deploy appropriate classification techniques and decision trees.
3. Understand the concept of clustering and its real time applications
4. Explore recent trends in data mining such as web mining, spatial-temporal mining
5. Able to know the basic concepts of data warehouse and OLAP operations
6. Organize and Prepare the data needed for data mining using pre preprocessing techniques

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. Using iris dataset preprocess and classify it with J4.8 and Naïve Bayes Classifier. examine the tree in the Classifier output panel
3. Using the datasets *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 J48, oneR and ID3 for vote dataset
6. Using Weka Experimenter perform comparison analysis of Naïve Bayes with different datasets
7. Apply ZeroR, OneR, and J48, to classify the Iris data in an experiment using 10 train and test runs, with 66% of the data used for training and 34% used for testing.
8. Using Weka Knowledge flow Sett up a flow to load an ARFF file (batch mode) and perform a cross-validation using J48 (WEKA's C4.5 implementation).

		Semester – V
17CAU513A	**MICROPROCESSOR PRACTICAL**	**4H – 2C**

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

Course Objectives:

Enable the student

- To become familiar with the architecture and the instruction set of an Intel microprocessor
- To illustrate the architecture of 8085 and 8086 microprocessors.
- To introduce the programming and interfacing techniques of 8086 microprocessor.
- To analyse the basic concepts and programming of 8051 microcontroller
- To understand the interfacing circuits for various applications of 8051 microcontroller.
- To introduce the architecture of advanced microprocessors and microcontrollers.

Course Outcomes:

Upon the completion of the course, the student will able to:

1. Design and implement programs on 8086, ARM, PIC. CO2 Design I/O circuits.
2. The program prepares students to successfully compete for employment in Electronics, Manufacturing and Embedded fields.
3. Design Memory Interfacing circuits.
4. Design and implement 8051 microcontroller based systems.
5. Describe the architecture and instruction set of ARM microcontroller
6. Assembly language programming will be studied as well as the design of various types of digital and analog interfaces

Assembly Language Programming

1. Addition of 8/16 bit and array of Data

2. Subtraction of 8/16 bit and array of Data

3. Multiplication of 8-bit Number

4. Division of 8-bit Number Write a program for 32-bit binary division and multiplication

5. Fill and Transfer an Array of Data

6. Ascending and Descending of an array

7. Data Transfer using Parallel Ports

8. Stepper Motor Interface

9. Traffic Light Interface

10. A/D and D/A Converter

Semester – V

17CAU513B **PC HARDWARE & TROUBLE SHOOTING PRACTICAL** **4H – 2C**

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

Course Objectives:

Enable the student

- to assemble/setup and upgrade personal computer systems
- perform installation, configuration,
- upgrading of hardware and software,
- install/connect associated peripherals
- diagnose in hardware and software and other peripheral equipment.
- Troubleshoot the problem

Course Outcomes:

Upon the completion of the course, the student will be able to

1. assemble/setup and upgrade personal computer systems
2. perform installation, configuration,
3. upgrading of hardware and software,
4. install/connect associated peripherals
5. diagnose in hardware and software and other peripheral equipment.
6. Troubleshoot the problem

List of Programs:

1. Identifying External Ports and Interfacing
2. Identifying PC cards and Interfacing.
3. Assembling of PC
4. Preventive Maintenance of a PC
5. Trouble Shooting of SMPS
6. Keyboard Servicing
7. Study of CRT
8. Communication and Bus Interfacing
9. Partitioning and Formatting Hard disks.
10. Installing System And Application Software

Semester – V

17CAU514A DIGITAL IMAGE PROCESSING PRACTICAL 4H – 2C

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

Course Objectives

- 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 apply knowledge of mathematics, science, and engineering
- To design and conduct experiments, as well as to analyze and interpret data

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

a. OR operation between two images
b. Calculate intersection of two images
c. Water Marking using EX-OR operation
d. 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
    c. To write and execute program for wavelet transform on given image and perform inverse wavelet transform to reconstruct image.
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.

Semester – V

17CAU514B OPEN SOURCE TECHNOLOGIES PRACTICAL 4H – 2C

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

Course Objectives:

Enable the student

- To introduce open source technology for development of web applications.
- For Study the problems with traditional commercial software.
- To understand open source scripting language for programming in web environment i.e. PHP.
- To study the open source management system and connection with database.
- To learn open source web server, software tools.
- To learn the open source ideals in order to apply those principles

Course Outcomes (COs)

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

1. Leaned the need of open source technology, open source development model, application of open sources, aspects of open source movement
2. The students will be aware about the problems with traditional commercial software.
3. The student will be familiar with basis syntax of PHP, common PHP scripts elements.
4. The student will be familiar with creating of the server side scripting using PHP, implement PHP database connectivity, perform operation on database and open source database management system.
5. The students will be familiar with Working of different web Servers.
6. The students will be aware about the software tool and process like Eclipse IDE, Selenium ID

Semester – V

17CAU514B OPEN SOURCE TECHNOLOGIES PRACTICAL 4H – 2C

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 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.

Semester – VI

17CAU601A **PHP PROGRAMMING** **3H – 3C**

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

Course Objectives
Enable the student

- To write basic PHP syntax using various operators.
- To write PHP scripts to handle HTML forms.
- To analyze different tasks using PHP functions.
- To understand the regular expressions in PHP.
- To learn array data structure using PHP scripts.
- To Create conditional structures

Course Outcomes (COs)
Upon completion of this course, students will be able to:

1. Write PHP scripts using operators to perform various functions
2. Design PHP scripts to handle HTML forms.
3. Implement different types of PHP functions.
4. Write regular expressions including modifiers, operators, and metacharacters.
5. Create PHP scripts using array.
6. Analyze and solve various database tasks using the PHP language.

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, scope of PHP -Basic Syntax, PHP variables and constants -Types of data in PHP , Expressions, scopes 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 -Scope of Function Global and Local

Semester – VI

17CAU601A PHP PROGRAMMING 3H – 3C

UNIT-IV

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

17CAU601B	UNIX / LINUX PROGRAMMING	Semester – VI 3H – 3C

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

Course Objectives

Enable the student to

- Understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.
- Understand how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions
- Understand how the operating system abstractions can be implemented
- Understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software
- 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 objectives, avoiding deadlocks, as well as security and protection.
- provide introduction to UNIX Operating System and its File System
- gain an understanding of important aspects related to the SHELL and the process
- develop the ability to formulate regular expressions and use them for pattern matching.

Course Outcome

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

1. Describe the architecture and features of UNIX Operating System and distinguish it from other Operating System
2. Demonstrate UNIX commands for file handling and process control
3. Write Regular expressions for pattern matching and apply them to various filters for a specific task
4. Analyze a given problem and apply requisite facets of SHELL programming in order to devise a SHELL script to solve the problem
5. Implement various file processing commands used in UNIX.
6. Construct various shell scripts for simple applications.

Unit-I

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

Semester – VI

17CAU602A CLOUD COMPUTING 4H – 4C

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

Course Objectives

Enable the student to

- Basics of cloud computing.
- Key concepts of virtualization.
- Different Cloud Computing services
- Cloud Implementation, Programming and Mobile cloud computing
- Key components of Amazon Web Services
- Cloud Backup and solution

Course Outcomes (COs)

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

1. Define Cloud Computing and memorize the different Cloud service and deployment models
2. Describe importance of virtualization along with their technologies.
3. Use and Examine different cloud computing services
4. Analyze the components of open stack & Google Cloud platform and understand Mobile Cloud Computing
5. Describe the key components of Amazon web Service
6. Design & develop backup strategies for cloud data based on features

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*,Wiley-India.
2. Dimitris N. Chorafas,(2010). Cloud Computing Strategies ,CRC Press.
3. Gautam Shroff, (2010). *Enterprise Cloud Computing Technology Architecture Application*s, Cambridge University Press.
4. Gautam Shroff, (2010). Enterprise Cloud Computing Technology Architecture Applications, Cambridge University Press.
5. Nikos Antonopoulos, Lee Gillam,(2012). *Cloud Computing: Principles, Systems and Applications*, Springer Publications.
6. Ronald L. Krutz, Russell Dean Vines,(2010). Cloud Security: A Comprehensive Guide to Secure Cloud Computing, , Wiley-India.
7. Toby Velte, Anthony Velte, Robert Elsenpeter,(2010). Cloud Computing, A Practical Approach ,McGraw Hills.

Websites

1. en.wikipedia.org/wiki/Cloud_computing
2. www.ibm.com/cloud-computing/in/en/
3. www.oracle.com/CloudComputing
4. www.microsoft.com/en-us/cloud/default.aspx

17CAU602B DATABASE ADMINISTRATION 4H - 4C

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

Course Objectives

Enable the student to

- Installing Oracle Software
- Creating an Oracle Database Using DBCA
- Managing Database instances and ASM instances
- Managing and controlling database network environment
- Define and devise transaction management, concurrency control, crash recovery components
- Managing storage structures

Course Outcomes (COs)

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

1. Design, model and install any database management systems by using Oracle database as sample.
2. Plan, design, construct, control and manage database instances, database network environment, storage structures, user security, database backup and recovery, database maintenance
3. Define and devise transaction management, concurrency control, crash recovery components
4. Examine and perform data base administration roles and operations by using Oracle database system as a sample
5. Compare and contrast by examining the database systems and new trends in data storage, data retrieval and maintenance techniques.
6. Configure, manage and maintain database, to audit and improve database performances and to use tools for database administration.

Unit I

Oracle DBA's: The Oracle DBA's Role- Oracle Database 11g Architecture: Oracle Databases and instances- Oracle Logical Storage structures – Oracle Logical Database structures – Oracle Physical Storage structures- Multiplexing Database Files - Oracle Memory Structures-Oracle Backup and Recovery – Security Capabilities – Tablespace Architecture – Oracle Tablespace installation – Traditional Disk Space Storage – Automatic Storage Management

Unit II

Common Space Management Problems – Oracle Segments, Extents and Blocks – Space Management Methodologies – SYSAUX monitoring and usage – Archived Redo Log

17CAU602B DATABASE ADMINISTRATION 4H - 4C

File Management – Built in Space Management Tools: Segment Advisor – Undo Advisor and the Automatic Workload Repository – Index usage – Space Usage Warning Levels – Reusable space allocation – Managing alert and Trace Files with ADR – Transaction Basics – Undo Basics – Managing Undo Tablespaces – Flashback features

Unit III

Tuning Application Design – Tuning SQL – Tuning Memory Usage – Tuning Data Access – Tuning Data Manipulation – Tuning Physical Storage – Reducing Network Security – Database Authentication Methods

Unit IV

Database Authorization Methods – Auditing: Auditing Locations – Statement Auditing – Privilege Auditing Schema Object Auditing – Auditing Related Data Dictionary Views – Logical Backups – Physical Backups – Using Data Pump Export and Import – Data Pump Import Options – Integration of Backup Procedures
Overview of Oracle Net – Using the Oracle Net Configuration Assistant – Using the Oracle Net Manager – Starting the Listener Server Process – Controlling the Listener Server Process Using Data links

Unit-V

Creating Tablespaces in a VLDB Environment: Bigfile Tablespace Basics – Creating and Modifying Bigfile Tablespace – Bigfile Tablespace ROWID format – DBMS_ROWID and Bigfile Tablespaces.- Advanced Oracle Table Types – Using Bitmap Indexes – Oracle Data Pump
Remote queries – Remote Data Manipulation: Two Phase Commit – Managing Distributed Data – Managing Distributed Transactions – Monitoring and Tuning Distributed Database

Suggested Readings

1. Bob Bryla, Kevin Loney 2008 Oracle Database 11g DBA Handbook McGraw-Hill Osborne
2. Saikat Basak. 2010. Oracle DBA Concise Handbook ,Ensel Software

Websites :

1. www.oracle.com/technology/software/products/database/oracle10g/index.html
2. www.oracle-base.com/articles/10g/
3. www.adp-gmbh.ch/ora/misc/10g.html

SEMESTER-VI

17CAU603A BIG DATA ANALYTICS 4H- 4C

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

End Semester Exam: 3Hours

Course Objectives

Enable the student

- To know the fundamental concepts of big data and analytics.
- To explore tools and practices for working with big data
- To learn about stream computing.
- To know about the research that requires the integration of large amounts of data.
- To introduce the tools required to manage and analyze big data like Hadoop, NoSql MapReduce.
- To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability

Course Outcomes (COs)

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

1. Work with big data tools and its analysis techniques
2. Analyze data by utilizing clustering and classification algorithms
3. Learn and apply different mining algorithms and recommendation systems for large volumes of data
4. Perform analytics on data streams
5. Learn NoSQL databases and management.
6. Understand the key issues in big data management and its associated applications in intelligent business and scientific computing

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.

17CAU603A BIG DATA ANALYTICS 4H- 4C

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
2. www.planet-data.eu/sites/default/files/Big_Data_Tutorial_part4.pdf
3. www.ibm.com/developerworks/data
4. www.solacesystems.com
5. en.wikipedia.org/wiki/Big_data
6. www.sap.com/solution/big-data.html

		Semester – VI
17CAU603B	**SYSTEM PROGRAMMING**	**4H – 4C**

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

Course Objectives

- 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.
- Describe the utility of different system programs & system tools.
- Familiarize with the tradeoffs between run-time and compile-time processing (Linking & Loading techniques).
- Explore the use of compiler with its phases.
- Use of Syntax directed scheme for intermediate code generation.
- Construct & use of different compiler tools as LeX, Yacc for code generation & optimization.

Course Outcomes

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

1. Organize the functionalities & components of system software & tools into different layers for efficient code generation.
2. Apply the knowledge & technique to develop solutions to real world problems by compiling application programs.
3. ability to identify, formulate, and solve computer engineering problems with proper systematic & semantic approach.
4. Develop possible program constructs for further code generation with Type checking & memory management strategy
5. Design a simple compiler with tools & different with optimized techniques
6. Design and implement system utility programs.

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, yacc. **Intermediate representations: Three** address code generation, syntax directed translation, translation of types, control Statements.

17CAU603B **SYSTEM PROGRAMMING** **Semester – VI**
4H – 4C

UNIT-IV

Storage organization: Activation records stack allocation.

UNIT-V

Code Generation: Object code generation

Suggested Readings

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

Semester – VI

17CAU611A PHP PROGRAMMING PRACTICAL 3H – 1C

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

Course Objectives
Enable the student

- To write basic PHP syntax using various operators.
- To write PHP scripts to handle HTML forms.
- To analyze different tasks using PHP functions.
- To understand the regular expressions in PHP.
- To learn array data structure using PHP scripts.
- To Create conditional structures

Course Outcomes (COs)

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

1. Write PHP scripts using operators to perform various functions
2. Design PHP scripts to handle HTML forms.
3. Implement different types of PHP functions.
4. Write regular expressions including modifiers, operators, and metacharacters.
5. Create PHP scripts using array.
6. Analyze and solve various database tasks using the PHP language.

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. Write a simple PHP program 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 : Thequick""brownfox

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. Write a simple PHP program 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,

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 FibonaCSUi 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.

Semester – VI

17CAU611B UNIX / LINUX PROGRAMMING PRACTICAL 3H – 1C

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

Course Objectives

Enable the student to

- Understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.
- Understand how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions
- Understand how the operating system abstractions can be implemented
- Understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software
- 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 objectives, avoiding deadlocks, as well as security and protection.
- provide introduction to UNIX Operating System and its File System
- gain an understanding of important aspects related to the SHELL and the process
- develop the ability to formulate regular expressions and use them for pattern matching.

Course Outcome

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

1. Describe the architecture and features of UNIX Operating System and distinguish it from other Operating System
2. Demonstrate UNIX commands for file handling and process control
3. Write Regular expressions for pattern matching and apply them to various filters for a specific task
4. Analyze a given problem and apply requisite facets of SHELL programming in order to devise a SHELL script to solve the problem
5. Implement various file processing commands used in UNIX.
6. Construct various shell scripts for simple applications.

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

		Semester – VI
17CAU612A	**CLOUD COMPUTING PRACTICAL**	**4H – 2C**

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

Course Objectives

Enable the student to

- Basics of cloud computing.
- Key concepts of virtualization.
- Different Cloud Computing services
- Cloud Implementation, Programming and Mobile cloud computing
- Key components of Amazon Web Services
- Cloud Backup and solution

Course Outcomes (COs)

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

1. Define Cloud Computing and memorize the different Cloud service and deployment models
2. Describe importance of virtualization along with their technologies.
3. Use and Examine different cloud computing services
4. Analyze the components of open stack & Google Cloud platform and understand Mobile Cloud Computing
5. Describe the key components of Amazon web Service
6. Design & develop backup strategies for cloud data based on features

List of Programs:

1. Create virtual machines that access different programs on same platform.

2. Create virtual machines that access 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

SEMESTER-VI

17CAU612B DATABASEADMINISTRATIONPRACTICAL 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

Enable the student to

- Installing Oracle Software
- Creating an Oracle Database Using DBCA
- Managing Database instances and ASM instances
- Managing and controlling database network environment
- Define and devise transaction management, concurrency control, crash recovery components
- Managing storage structures
- Controlling user security
- Designing Database backup and recovery procedures
- Take Decisions related with Database Maintenance

Course Outcomes (COs)

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

1. Design, model and install any database management systems by using Oracle database as sample.
2. Plan, design, construct, control and manage database instances, database network environment, storage structures, user security, database backup and recovery, database maintenance
3. Define and devise transaction management, concurrency control, crash recovery components
4. Examine and perform data base administration roles and operations by using Oracle database system as a sample
5. Compare and contrast by examining the database systems and new trends in data storage, data retrieval and maintenance techniques.
6. Configure, manage and maintain database, to audit and improve database performances and to use tools for database administration.

List of Programs:

1. Demo for Globalization Support
2. Setup Listener Security
3. Configuring Recovery Manager
4. Write a program Using Recovery Manager
5. Write a program for Managing Diagnostic Sources
6. Implement Database Recovery

Semester-VI

17CAU612B DATABASEADMINISTRATIONPRACTICAL 4H-2C

7. Demo for Flashback Database
8. Implement User Error Recovery
9. Write a program for Dealing with Corruption
10. Show the demo for Automated Management
11. Creating a database and do the manipulation.
12. Managing index tables

Semester – VI

17CAU613A BIG DATA ANALYTIC PRACTICAL 4H – 2C

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

Course Objectives

Enable the student

- To know the fundamental concepts of big data and analytics.
- To explore tools and practices for working with big data
- To learn about stream computing.
- To know about the research that requires the integration of large amounts of data.
- To introduce the tools required to manage and analyze big data like Hadoop, NoSql MapReduce.
- To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability

Course Outcomes (COs)

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

1. Work with big data tools and its analysis techniques
2. Analyze data by utilizing clustering and classification algorithms
3. Learn and apply different mining algorithms and recommendation systems for large volumes of data
4. Perform analytics on data streams
5. Learn NoSQL databases and management.
6. Understand the key issues in big data management and its associated applications in intelligent business and scientific computing

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.

Semester – VI

17CAU613B SYSTEM PROGRAMMING PRACTICAL 4H – 2C

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

Course Objectives

- 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.
- Describe the utility of different system programs & system tools.
- Familiarize with the tradeoffs between run-time and compile-time processing (Linking & Loading techniques).
- Explore the use of compiler with its phases.
- Use of Syntax directed scheme for intermediate code generation.
- Construct & use of different compiler tools as LeX, Yacc for code generation & optimization.

Course Outcome

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

1. Organize the functionalities & components of system software & tools into different layers for efficient code generation.
2. Apply the knowledge & technique to develop solutions to real world problems by compiling application programs.
3. ability to identify, formulate, and solve computer engineering problems with proper systematic & semantic approach.
4. Develop possible program constructs for further code generation with Type checking & memory management strategy
5. Design a simple compiler with tools & different with optimized techniques
6. Design and implement system utility programs.

List of Programs:

1. To implement an assembler for a hypothetical language.

2. To get familiar with lex: write a program to recognize numbers, identifiers.

3. To get familiar with yaCSU: write a desk calculator.