

பகுதி - I தமிழ்ப் பாடத்திட்டம் (2016 - 2017)
முதல்பருவம்
(இளநிலை அறிவியல் பட்ட வகுப்புகளுக்குரியது)
(For I-UG Science Degree Classes) 16LSU101

பாடத்திட்டப் பொதுநோக்கம்

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

பாடத்திட்டப் பயன் விளைவு

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

பகுதி - I தமிழ்ப் பாடத்திட்டம் (2016-2017)

பகுதி - I, தமிழ்

பருவம் I

16LAU101 :

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, வினைத்திட்டம் - 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. படைப்பிலக்கியப் பயிற்சிகள் (கதை, கவிதை, கட்டுரை, உரைநடை)

2. மொழிபெயர்ப்பு

3. இலக்கணப் பயிற்சிகள்

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

Instruction Hours / week: L: 4 T: 0 P: 0

Marks: Internal: 100

Total: 100

Course Objectives (CO)

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

Course Outcomes (COs)

After the completion of this course, a successful 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. III-effects of fire works.

Unit III: Biodiversity and Its Conservation: Introduction, definition: genetic, species and ecosystem diversity. Biogeographical classification of India. Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values. Biodiversity at global, National and local levels. India as a mega-diversity nation. Hot-spots of

biodiversity. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts. Endangered and endemic species of India. Conservation of biodiversity: in-situ and ex-situ conservation of biodiversity.

Unit IV: Environmental Pollution - Definition, Causes, effects and control measures of Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, Nuclear hazards, Solid waste management: Causes, effects and control measures of urban and industrial wastes. Role of an individual in prevention of pollution. Pollution case studies. Disaster management: Floods, 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

T1: Tripathy. S.N., & Sunakar Panda. (2004). Fundamentals of Environmental Studies (2nd ed.) . New Delhi: Vrianda Publications Private Ltd.

T2: Arvind Kumar. (2004). A Textbook of Environmental Science. New Delhi: APH Publishing Corporation.

T3: Verma, P.S., & Agarwal V.K. (2001). Environmental Biology (Principles of Ecology) . New Delhi: S.Chand and Company Ltd.

T4: Anubha Kaushik., & Kaushik, C.P. (2004). Perspectives in Environmental Studies. New Delhi: New Age International Pvt. Ltd. Publications.

R1: Singh, M.P., Singh, B.S., & Soma, S. Dey. (2004). Conservation of Biodiversity and Natural Resources. New Delhi: Daya Publishing House.

R2: Daniel, B. Botkin., & Edward, A. Keller. (1995). Environmental Science New York: John Wiley and Sons, Inc.,.

R3: Uberoi, N.K. (2005). Environmental Studies. New Delhi: Excel Books Publications.

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT-I**Introduction to C and C++:**

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

Data Types, Variables, Constants, Operators and Basic I/O: Declaration, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators, Using Comments in programs, Character I/O, Formatted and Console I/O, Using Basic Header Files.

Expressions, Conditional Statements and Iterative Statements: Simple Expressions in C++, Understanding Operators Precedence in Expressions, Conditional Statements, Understanding syntax and utility of Iterative Statements, Use of break and continue in Loops, Using Nested Statements.

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, Use Various types of arrays, Two-dimensional Arrays, 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, Pointers to Pointers, Pointers to structures, Passing pointers as function arguments, Returning a pointer from a function, using arrays as pointers, Passing arrays to functions.

UNIT-IV

File I/O, Preprocessor Directives: Opening and closing a file, Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives, Macros.

UNIT-V

Using Classes in C++: Principles of Object-Oriented Programming, Defining & Using Classes, Constructors, Constructor Overloading, Function overloading and Operator overloading in classes, Class Variables & Functions, Access specifiers, Overview of Template classes and their use.

Inheritance, Polymorphism and Exception Handling: Introduction to Inheritance, Polymorphism, Basics Exceptional Handling.

Suggested Readings:

1. BjarneStroustrup, "Programming -- Principles and Practice using C++", 2nd Edition, Addison-Wesley 2014.
2. BjarneStroustrup, "The C++ Programming Language", 4th Edition, Addison-Wesley, 2013

3. Harry, H. Chaudhary, "Head First C++ Programming: The Definitive Beginner's Guide", First Create space Inc, O-D Publishing, LLC USA.2014
4. Stanley B. Lippman, JoseeLajoie, Barbara E. Moo, "C++ Primer", Published by Addison-Wesley, 5th Edition, 2012
5. Paul Deitel, Harvey Deitel, "C++ How to Program", 8th Edition, Prentice Hall, 2011.
6. E Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw-Hill Education, 2008.
7. Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.
- 8 . Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.
9. Scott Meyers, "Effective C++", 3rd Edition, Published by Addison-Wesley, 2005.

WEB SITES

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT -I Introduction

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

UNIT -II Data Representation and Basic Computer Arithmetic

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

UNIT –III Basic Computer Organization and Design

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

UNIT-IV Central Processing Unit

Register organization, arithmetic and logical micro-operations, stack organization, micro programmed control. Instruction formats, addressing modes, instruction codes, machine language, assembly language, input output programming, RISC, CISC architectures, pipelining and parallel architecture.

UNIT –V Memory and Input-Output Organization

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

Suggested Readings:

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

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

Course Objectives (CO)

- To develop a basic understanding of many areas of information technology and how they are used
- To understand the basic structure of a program including sequence, decisions and looping
- To understand how to design a program to solve a simple program
- To introduce the basics of several programming language and understand the commonality and differences in languages
- To lay a basic foundation involving hardware, software, navigation, the Internet for future development
- To understand what a database is and how to design a working model

Course Outcomes (COs)

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

1. Demonstrate knowledge of high-level structured programming logic and algorithm development.
2. Demonstrate knowledge of structured program design and modularity.
3. Demonstrate knowledge of file-based input/output operations, file manipulation and maintenance, extract reporting, and report writing.
4. Demonstrate knowledge of records, data types and structures, storage classes, addressable memory locations.
5. Demonstrate knowledge of table utilization, arrays and subscripts.
6. Demonstrate knowledge of arithmetic expressions, control structures, iteration techniques.

UNIT – I

Language Evolution Machine Language, Assembly Language, High Level Language. Translators: compiler, Interpreter and Assembler. The Compilation Process, Linker, Loader, Study of HLL, Characteristics of Good Language, Generation of Languages, Study of Programming Languages (Function Oriented, Object Oriented, Event-Based).

UNIT – II

Programming Construction Tools Problem Analysis, Process Analysis, Conceptual Development of Solution. Development Tools: Algorithm: Types of Algorithm, Algorithm of Analysis, Advantage and Disadvantage of Algorithm, Complexity of Algorithm, Big-O Notation.

UNIT – III

Flowcharts: Types of Flowcharts, Advantage and Disadvantage of Flowchart.
Pseudocode: Definition and Its Characteristics Control Statements Basics of
Programming Language: Usage of Character Set, Meaning of Keywords and Identifiers,
Role of Data Types, Constants and Variables.

UNIT – IV

Importance of Casting, Different Types of Operators and their Precedence, Expressions,
Conditional Statements (One-Way, Two-Way and Multi-Way Conditional), Looping
Statements (For, While, do-while), Usage of Exit, continue, Break and Goto Statement.

UNIT – V

Arrays: Arrays, One dimensional array, Various operation on Array (Inserting of Elements,
Deleting of Element, Rotating List, Sorting, Searching, Merging Etc) and Two
dimensional arrays (Matrix Addition, Transpose of Matrix, Matrix Multiplication),
Modular programming and its features.

Suggested Readings:

1. Anil V.Chouduri. The Art of Programming through Flowchart and Algorithms. Laxmi Publication.
2. Maureen Sprankle (2009). Problem Solving Programming Concepts (7th ed.). Pearson Education.
3. Behrouz Forouzan. Basic of Computer Science. Cengage Learning
4. Donald Knuth. The Art of Computer Programming Vol-I,II, III, Pearson.
 5. Horowitz, Sahani. Fundamental of Computer Algorithm. Orient Longman.

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

List of Programs

1. WAP to display Fibonacci series (i)using recursion, (ii) using iteration
2. WAP to compute the sum of the first n terms of the following series
 $S = 1 + 1/2 + 1/3 + 1/4 + \dots$
3. Write a program to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.

4. WAP to print the sum and product of digits of an integer.
5. WAP to reverse a number.
6. Write a program that checks whether a given string is Palindrome or not.
7. WAP to print a triangle of stars as follows (take number of lines from user):

*

8. WAP to perform following actions on an array entered by the user:
 - i) Calculate and print the sum and average of the elements of array
 - ii) Print the maximum and minimum element of array
9. Write a program that swaps two numbers using pointers.
10. 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.
11. Write a program to perform operations on strings.
12. Write a program to perform following Matrix operations (2-D array implementation):
 - a) Sum b) Difference c) Product
13. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
14. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
15. Copy the contents of one text file to another file using command line argument.

16CTU112 COMPUTER SYSTEM ARCHITECTURE- PRACTICAL 3H – 2C

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

List of Experiments

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

List of Programs

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

MS Word

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

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

- Leave a gap of 12-points after the title.

2. Create a **telephone directory**.

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

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

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

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

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

5. Create the following one page documents.

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

6. Create the following documents:

- (a) A newsletter with a headline and 2 columns in portrait orientation, including at least one image surrounded by text.
- (b) Use a newsletter format to promote upcoming projects or events in your classroom or college.

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

Type the following as shown (do not bold).

Color, Style, Item

Blue, A980, Van

Red, X023, Car

Green, YL724, Truck

Name, Age, Sex

Bob, 23, M

Linda, 46, F

Tom, 29, M

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

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

Add a column Region (values: S, N, N,S,S,S) between the Salesperson and Dolls columns to the given table Sort your table data by Region and within Region by Salesperson in ascending order:

In this exercise, you will add a new row to your table, place the word Total at the bottom of the Salesperson column, and sum the Dolls, Trucks, and Puzzles columns.

9. Wrapping of text around the image.

10. Following features of menu option must be covered

FILE Complete menu

EDIT Complete menu

VIEW Complete menu

INSERT Complete menu

FORMAT Complete menu

TABLE Complete menu

WINDOW Complete menu

HELP Complete menu

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

MS Excel

1. Enter the Following data in Excel Sheet

REGIONAL SALES PROJECTION

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

TOTAL AVERAGE

(a) Apply Formatting as follow:

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

(b) Calculate State and Qtr. Total

(c) Calculate Average for each quarter

(d) Calculate Amount = Rate * Total .

2. Given the following worksheet

	A	B	C	D
1	Roll No.	Name	Marks	Grade
2	1001	Sachin	99	
3	1002	Sehwag	65	
4	1003	Rahul	41	
5	1004	Sourav	89	
6	1005	Har Bhajan	56	

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

If Marks	Then Grade
>= 80	A+
>= 60 < 80	A

>= 50 < 60	B
< 50	F

3. Given the following worksheet

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

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

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

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

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

Allowances

- HRA Dependent on Basic
 - 30% of Basic if Basic <=1000
 - 25% of Basic if Basic>1000 & Basic<=3000
 - 20% of Basic if Basic >3000
- DA Fixed for all employees, 30% of Basic
- Conveyance Allowance Rs. 50/- if Basic is <=1000
Rs. 75/- if Basic >1000 & Basic<=2000
Rs. 100 if Basic >2000
- Entertainment Allowance NIL if Basic is <=1000 Rs. 100/- if Basic > 1000

Deductions

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

Calculate the following:

Gross Salary = Basic + HRA + DA + Conveyance + Entertainment
 Total deduction = Provident Fund + Group Insurance Premium
 Net Salary = Gross Salary – Total Deduction.

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

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

6. Use an array formula to calculate Simple Interest for given principal amount. Given the Rate of Interest and time.

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

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

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

- Calculate total sale year wise.
- Calculate the net sale made by each salesman
- Calculate the maximum sale made by the salesman
- Calculate the commission for each salesman under the condition.
 - If total sales >4,00,000 give 5% commission on total sale made by the salesman.
 - Otherwise give 2% commission.
- Draw a bar graph representing the sale made by each salesman.
- Draw a pie graph representing the sale made by salesman in 2000.

8. Enter the following data in Excel Sheet

PERSONAL BUDGET FOR FIRST QUARTER
Monthly Income (Net): 1,475

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

Monthly Total

Calculate Quarter total and Quarter average.

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

9. Enter the following data in Excel Sheet

TOTAL REVENUE EARNED FOR SAM'S BOOKSTALL

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

- Compute the total revenue earned.
- Plot the line chart to compare the revenue of all publisher for 4 years.
- Chart Title should be _Total Revenue of sam's Bookstall (1997-2000)'
- Give appropriate categories and value axis title.

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

பகுதி - I தமிழ்ப் பாடத்திட்டம் (2016 - 2017)
இரண்டாம் பருவம்
(இளநிலை அறிவியல் பட்ட வகுப்புகளுக்குரியது)
(For I-UG Science Degree Classes) 16LSU201

பாடத்திட்டப் பொதுநோக்கம்

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

பாடத்திட்டப் பயன் விளைவு

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

பகுதி - I தமிழ்ப் பாடத்திட்டம் (2016 - 2017)

பகுதி - I, தமிழ்

பருவம் II

16LAU201 :

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 நகாச்சிறப்பு -

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

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

கலித்தொகை : சுடாத்தொட 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.

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

WEB SITES

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT I

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

UNIT II

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

UNIT III

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

UNIT IV

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

UNIT V

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

Suggested Readings

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

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

Course Objectives (CO)

- To understand the application development environment.
- To gain programming Skills and Database Creation in RDBMS.
- To understand SQL for storing and retrieving data from the RDBMS.
- To arrive at a normalized design of tables and other database objects in RDBMS.
- To investigate techniques for tuning your PL/SQL code
- To design modular applications using packages

Course Outcomes (COs)

After completion of this course, the students will be 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. Invoke native dynamic SQL to build runtime SQL statements
4. Manage data retrieval with cursors and cursor variables
5. Manage dependency issues and collect PL/Scope data
6. Create triggers to solve business challenges and enforce business rules

UNIT - I Understanding Database Fundamentals

Origin of database – database elements – design concepts – components of DBMS – Advantages and disadvantages of DBMS. Database Models: flat file – hierarchical model – network model – relational model – object oriented model – Features of Object oriented Database Management system – Features of distributed DBMS – Comparison of DBMS & DDBMS – Object relational model. ER-model: entities – relationships - ERD symbols – cardinalities – sample ERD.

Unit - II Entities and Entity Relationships

Relational model: Introduction – Relational database: attributes and domain – Tuples – Relation and their schemes – Relation representation – keys – relationships – relational operations – Integrity rules. Relational algebra: Basic operations – Additional relational algebraic operations – some relational algebra queries. Functional Dependency: Reasoning about FD's – closure of set of FD's – Attribute closure.

Unit - III Objects

Relational database manipulation: Introduction – SQL: Data definition – Data manipulation: Basic data retrieval – condition specification – Arithmetic and aggregate operations. SQL joins – set manipulation – categorization – updates – views – index. Data Control language : grant, revoke – simple privileges.

Unit - IV Overview of PL/SQL

Declaration section – executable command section : conditional logic, loops, CASE statements – exception handling section: predefined and user defined exceptions. Triggers : definition – types : row level, statement level, before and after, instead of – syntax – enabling and disabling triggers - replacing and dropping triggers. Cursors – definition – open – fetch – close – cursor attributes- select for update – types : implicit, explicit. Procedures, Functions: Local and global – procedures vs functions – stored procedures, functions – create procedure syntax - create function syntax – calling procedures, functions. Replacing and dropping procedures, functions.

Unit - V Packages and Normalization

Package header – package body – calling package members - Replacing and dropping package. Overview of Normalization : advantages - disadvantages. Normal forms: first normal form – second normal form – third normal form – boyce codd normal form – Introduction to fourth, fifth and sixth normal forms – denormalization. Parallel Databases: Introduction – Design of Parallel Databases – Advantages and Disadvantages of Parallel Database.

Suggested Readings

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

Web Sites

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

List of Programs

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

10. Modify the `Distance` class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions (from lower to higher data type)
12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
13. Write a program to show the use of static functions and to pass variable length arguments in a function.
14. Write a program to demonstrate the concept of boxing and unboxing.
15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate fibonacci series is given in a different file belonging to the same package.
17. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
18. Write a program `DivideByZero` that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
20. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
21. Write a program to demonstrate priorities among multiple threads.
22. Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).
23. Write a program to create URL object, create a URLConnection using the `openConnection()` method and then use it to examine the different components of the URL and content.
24. Write a program to implement a simple datagram client and server in which a message that is typed into the server window is sent to the client side where it is displayed.
25. Write a program that creates a Banner and then creates a thread to scroll the message in the banner from left to right across the applet's window.
26. Write a program to get the URL/location of code (i.e. java code) and document (i.e. html file).
27. Write a program to demonstrate different mouse handling events like `MouseClicked()`, `MouseEntered()`, `MouseExited()`, `MousePressed()`, `MouseReleased()` and `MouseDragged()`.
28. Write a program to demonstrate different keyboard handling events.

29. Write a program to generate a window without an applet window using main() function.
30. Write a program to demonstrate the use of push buttons.

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

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

Course Outcomes (COs)

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

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

List of Programs

1. Write a C Program to find the number of subsets of a set contains n elements.
2. Write a C Program to find transitive closure of a relation.
3. Write a C Program to prove

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

**16CTU213 DATABASE PROGRAMMING WITH ORACLE
(SQL AND PL/SQL) - PRACTICAL**

**Semester – III
4H – 4C**

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

Course Objectives (CO)

- To understand the application development environment.
- To gain programming Skills and Database Creation in RDBMS.
- To understand SQL for storing and retrieving data from the RDBMS.
- To arrive at a normalized design of tables and other database objects in RDBMS.
- To investigate techniques for tuning your PL/SQL code
- To design modular applications using packages

Course Outcomes (COs)

After completion of this course, the students will be 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. Invoke native dynamic SQL to build runtime SQL statements
4. Manage data retrieval with cursors and cursor variables
5. Manage dependency issues and collect PL/Scope data
6. Create triggers to solve business challenges and enforce business rules

List of Programs

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		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, 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 Employee Name concatenated by a job separated by a comma.
4. Query to display Employee Name and salary of all the employees earning more than \$2850.
5. Query to display Employee Name and department number for the employee no=7900.
6. Query to display Employee Name and salary for all employees whose salary is not in the range of \$1500 and \$2850
7. Query to display Employee Name and department no. of all the employees in dept 10 and Dept 30 in the alphabetical order by name.
8. Query to display Name and Hire date of every employee who was hired in 1981
9. Query to display Name and job of all employees who don't have a current manager.
10. Query to display Name, salary and commission for all the employees who earn commission
11. Sort the data in descending order of salary and commission
12. Query to display Name of all the employees where the third letter of their name is A.
13. Query to display Name, salary and commission for all employees whose commission is 14 greater than their salary increased by 5%.
14. Query to display the current date
15. Query to display Name, hire date and salary review date which is the first Monday after 6 months of employment.
16. Query to display Name and calculate the number of months between today and the date each employee was hired.
17. Query to display the following for each employee <E-Name> earns <Salary> monthly but wants <3 * current salary>. Label the column as Dream salary
18. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all employees whose name starts with 'J', 'A' and 'M'.
19. Query to display unique listing of all jobs that are in dept #30.
20. 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.
21. Query to display the department name, location name, number of employees and the average salary for all the employees in that department.
22. Query to display employee no. and name for all employees who earn more than the average salary.

PL/SQL

1. Write the PL/SQL program to find the factorial and fibonacci series of given numbers.
- 2.(i) Write the PL/SQL program to check whether the string is Palindrome.
(ii) Write the PL/SQL program to reverse a number
(iii) Write the PL/SQL program to check whether the number is Armstrong

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

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

4. Create table with following fields:

Product table:

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

Vendor table:

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

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

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

Salary:

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

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

7. Create a table to contain phone_number, user_name, address. Write a function to search for address using phone_number.

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

(For all undergraduate students admitted from 2016 onwards)

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. Chandralekhya et al. (Spring 2013). Chennai: Emerald Publishers.

Suggested Reading:

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT-I

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

UNIT-II

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

UNIT-III

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

UNIT-IV

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

Techniques

UNIT-V

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

Suggested Readings

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

Web Sites

http://en.wikipedia.org/wiki/Data_structure

<http://www.cs.sunysb.edu/~skiena/214/lectures/>

www.amazon.com/Teach-Yourself-Structures-Algorithms

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

Course Objectives (CO)

- Understand the benefits of using Microsoft Visual Basic 6.0 for Windows as an application tool.
- Understand the Visual Basic event-driven programming concepts, terminology, and available tools.
- Learn the fundamentals of designing, implementing, and distributing a Visual Basic application.
- Learn to use the Visual Basic toolbox. Learn to modify object properties. Learn object methods.
- Use the menu design window. Understand proper debugging and error-handling procedures.
- Gain a basic understanding of database access and management using data bound controls. Obtain an introduction to ActiveX controls and the Windows Application Programming Interface (API).

Course Outcomes (COs)

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

1. Demonstrate knowledge of programming terminology and how applied using Visual Basic (e.g., variables, selection statements, repetition statements, etc.)
2. Develop a Graphical User Interface (GUI) based on problem description
3. Develop an Event Planning Chart based on problem description so as to define the processing that is to occur based on specific events
4. Develop an Algorithm to verify processing is accurate
5. Develop and debug applications using Visual Basic 2010 (or version required for the course) that runs under Windows operating system
6. Develop programs that retrieve input from a file as opposed to input only provided by user

UNIT I

Beginning Visual Basic: Introduction to Visual Basic: Introduction Graphics User Interface (GUI), Programming Language (Procedural, Object oriented, event driven), The Visual Basic environment IDE, Introduction to VB Controls: Textboxes, Frames, check boxes, options buttons, setting a border and style, the shape control, the line control, working with multiple controls and their properties, designing the user interface, keyboard access, tab controls. Default & controls property, coding for Controls, list box and combo box and their properties, filling the list box using property window/add item method, picture/image box and their properties.

UNIT II

Dealing with data: Operators-Variables-declaring variables- types of variables – data types – constants – arrays – declaring arrays – specifying arrays – Multidimensional arrays – dynamic arrays – arrays of arrays. Val function, Arithmetic operations, formatting data. Error functions and types. Introducing to Menu editor.

UNIT III

Writing Code: Control flow statements – If – Then – If-then-else – Nested control statements – Select case – Loop statements – Do-loop – For-Next – While Wend – Exit statement . Displaying message in Message box, testing whether input is valid or not. Collections – procedures – Subroutines – Functions – Calling procedures – Object Browser – Creating classes and Objects – I/O statements

UNIT IV

Working with forms and procedures: Introducing to forms and types of forms and setting form properties, creating, adding, removing forms in project, hide, show method, load, unload, statement, Me keywords, Referring to objects on a different forms. Creating an application using controls: What is on the toolbar – Textbox control – Picture box – Image box – Label box – Frame – List box – Option button – Combo box – Command Button – check box – The Drive, Directory, File list controls – The Line & Shape control – Scroll Box – Data – Timer.

UNIT V

Multiple Document Interface & Menus: Why MDI Forms – Features of an MDI forms – Loading MDI forms & child forms – creating a simple MDI forms –Accessing MDI forms – creating MENUS – POP-UP MENUS.

Data access controls: JET database Engine – ADODC – DAO Data control – ODBC Data Source Administrator – DATA REPORT.

Suggested Readings

1. Noel Jerke (2008). Visual Basic 6.0: The Complete Reference. Tata McGraw Hill Publishing Company Ltd.
2. Mohammed Azam. Programming with VB 6.0. Vikas Publishing.
3. Peter Wrights (1999). Beginning VB 6.0 (4th ed.). New York:Springer-Verlag Incorporated.

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT I

Introduction to 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-rate limits; 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; packet switching - connectionless datagram switching, connection-oriented virtual circuit switching; dial-up modems; digital subscriber line; cable TV for data transfer.

UNIT – III

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

UNIT – IV

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

Unit V

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

Suggested Readings

2. B. A Forouzan (2007). Data Communications and Networking(4th ed.). THM.
3. A. S.Tanenbaum, (2002). Computer Networks (4th ed.). PHI.

WEB SITES

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

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

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

Course Outcomes (COs)

Upon completion of this course, the students will able to

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

UNIT-I

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

UNIT-II

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

UNIT-III

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

UNIT-IV

User Interface Architecture: Application context, intents, Activity life cycle, multiple screen size s.

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

UNIT-V

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

Suggested Readings

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

REFERENCES

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

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

Course Objectives (CO)

- To familiarize of the syntax, semantics, data-types and library functions of numerical computing languages such as MATLAB and/or SCILAB
- To reinforce a structured, top-down approach to formulate and solve problems.
- To introduce common approaches, structures, and conventions for creating and evaluating computer programs, primarily in a procedural paradigm, but with a brief introduction to object-oriented concepts and terminology.
- To understand a variety of common numeric techniques to solve and visualize engineering-related computational problems.
- To introduce the MATLAB software environment.
- To implement/simulate and visualization of basic mathematical functions relevant to electronics applications.

Course Outcomes (Cos)

By the end of this course, students should be able to

1. Use MATLAB effectively to analyze and visualize data.
2. Apply numeric techniques and computer simulations to solve engineering-related problems.
3. Apply a top-down, modular, and systematic approach to design, write, test, and debug sequential MATLAB programs to achieve computational objectives.
4. Design and document computer programs and analyses in a careful and complete manner so as to effectively communicate results, to facilitate evaluation and debugging by another programmer, and to anticipate and resolve user errors.
5. Demonstrate understanding and use of fundamental data structures (classes).
6. Create and control simple plot and user-interface graphics objects in MATLAB.

UNIT-I

Introduction to Programming: Components of a computer, working with numbers, Machine code, Software hierarchy.

UNIT-II

Programming Environment: MATLAB Windows, A First Program, Expressions, Constants, Variables and assignment statement, Arrays.

UNIT-III

Graph Plots: Basic plotting, Built in functions, Generating waveforms, Sound replay, load and save. Procedures and Functions: Arguments and return values, M-files, Formatted console input-output, String handling,

UNIT-IV

Control Statements: Conditional statements: If, Else, Else-if, Repetition statements: While, for loop

UNIT-V

Manipulating Text: Writing to a text file, Reading from a text file, Randomising and sorting a list, searching a list. **GUI Interface:** Attaching buttons to actions, Getting Input, Setting Output

Suggested Readings

1. Amos Gilat (2004). MATLAB: An Introduction with Applications(2nd ed). New Delhi: Wiley.
2. C.B. Moler (2004). Numerical Computing with MATLAB. SIAM.

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

Total: 100

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

List of Programs

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

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

16CTU312 PROGRAMMING WITH VISUAL BASIC - PRACTICAL 4H – 2C

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

Course Objectives (CO)

- Understand the benefits of using Microsoft Visual Basic 6.0 for Windows as an application tool.
- Understand the Visual Basic event-driven programming concepts, terminology, and available tools.
- Learn the fundamentals of designing, implementing, and distributing a Visual Basic application.
- Learn to use the Visual Basic toolbox. Learn to modify object properties. Learn object methods.
- Use the menu design window. Understand proper debugging and error-handling procedures.
- Gain a basic understanding of database access and management using data bound controls. Obtain an introduction to ActiveX controls and the Windows Application Programming Interface (API).

Course Outcomes (COs)

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

1. Demonstrate knowledge of programming terminology and how applied using Visual Basic (e.g., variables, selection statements, repetition statements, etc.)
2. Develop a Graphical User Interface (GUI) based on problem description
3. Develop an Event Planning Chart based on problem description so as to define the processing that is to occur based on specific events
4. Develop an Algorithm to verify processing is accurate
5. Develop and debug applications using Visual Basic 2010 (or version required for the course) that runs under Windows operating system
6. Develop programs that retrieve input from a file as opposed to input only provided by user

List of Programs

Implement the following projects using Visual Basic Language

1. To calculate the Simple interest and compound interest
2. To perform the following string functions
 - a. Upper to lower
 - b. String concatenation
 - c. String length

- d. String compare
 - e. Bold, Italic and Underline
 - f. Rtrim and Ltrim
 - g. Change the background of form
3. To draw different shapes and fill with different colors using MDI form
 4. To implement a simple calculator
 5. To create an ActiveX Control
 6. To animate the picture using timer control
 7. To convert text to voice
 8. To simulate web browser
 9. Develop a project for Railway Reservation System
 10. Develop a project for Employee Payroll system
 11. Develop a project for Student Information system. Generate data report for Student marklist
 12. Develop a project to generate Barcode for Library applications

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

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

Course Outcomes (COs)

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

List of Programs

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

Course Objectives (CO)

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

Course Outcomes (COs)

Upon completion of this course, the students will able to

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

List of Programs

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

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

- To familiarize of the syntax, semantics, data-types and library functions of numerical computing languages such as MATLAB and/or SCILAB
- To reinforce a structured, top-down approach to formulate and solve problems.
- To introduce common approaches, structures, and conventions for creating and evaluating computer programs, primarily in a procedural paradigm, but with a brief introduction to object-oriented concepts and terminology.
- To understand a variety of common numeric techniques to solve and visualize engineering-related computational problems.
- To introduce the MATLAB software environment.
- To implement/simulate and visualization of basic mathematical functions relevant to electronics applications.

Course Outcomes (Cos)

By the end of this course, students should be able to

1. Use MATLAB effectively to analyze and visualize data.
2. Apply numeric techniques and computer simulations to solve engineering-related problems.
3. Apply a top-down, modular, and systematic approach to design, write, test, and debug sequential MATLAB programs to achieve computational objectives.
4. Design and document computer programs and analyses in a careful and complete manner so as to effectively communicate results, to facilitate evaluation and debugging by another programmer, and to anticipate and resolve user errors.
5. Demonstrate understanding and use of fundamental data structures (classes).
6. Create and control simple plot and user-interface graphics objects in MATLAB.

List of Programs

1. Write a program to assign the following expressions to a variable A and then to print out the value of A.
 - a. $(3+4)/(5+6)$
 - b. $2\pi^2$
 - c. $\sqrt{2}$
 - d. $(0.0000123 + 5.67 \times 10^{-3}) \times 0.4567 \times 10^{-4}$
2. Celsius temperatures can be converted to Fahrenheit by multiplying by 9, dividing by 5, and adding 32. Assign a variable called C the value 37, and implement this formula to assign a variable F the Fahrenheit equivalent of 37 Celsius.
3. Set up a vector called N with five elements having the values: 1, 2, 3, 4, 5. Using N, create assignment statements for a vector X which will result in X having these values:
 - a. 2, 4, 6, 8, 10
 - b. $1/2$, 1, $3/2$, 2, $5/2$

- c. 1, 1/2, 1/3, 1/4, 1/5
d. 1, 1/4, 1/9, 1/16, 1/25
4. A supermarket conveyor belt holds an array of groceries. The price of each product (in pounds) is [0.6, 1.2, 0.5, 1.3] ; while the numbers of each product are [3, 2, 1, 5]. Use MATLAB to calculate the total bill.
5. The `sortrows(x)` function will sort a vector or matrix X into increasing row order. Use this function to sort a list of names into alphabetical order.
6. The —identityl matrix is a square matrix that has ones on the diagonal and zeros elsewhere. You can generate one with the `eye()` function in MATLAB. Use MATLAB to find a matrix B, such that when multiplied by matrix $A = \begin{bmatrix} 1 & 2 \\ -1 & 0 \end{bmatrix}$ the identity matrix $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is generated. That is $A*B = I$.
7. Create an array of N numbers. Now find a single MATLAB statement that picks out from that array the 1,4,9,16,...,√Nth entries, i.e. those numbers which have indices that are square numbers.
8. Draw a graph that joins the points (0,1), (4,3), (2,0) and (5,-2).
9. The seeds on a sunflower are distributed according to the formula below. Plot a small circle at each of the first 1000 co-ordinates :
- $$r_n = \sqrt{n}$$
- $$\theta_n = \frac{137.51}{180} \pi n$$
10. Calculate 10 approximate points from the function $y=2x$ by using the formulae:
i. $x_n = n$
ii. $y_n = 2n + \text{rand} - 0.5$
Fit a line of best fit to these points using the function `polyfit()` with `degree=1`, and generate co-ordinates from the line of best fit using `polyval()`. Use the on-line help to find out how to use these functions. Plot the raw data and the line of best fit.
11. Calculate and replay 1 second of a sinewave at 500Hz with a sampling rate of 11025Hz. Save the sound to a file called `ex35.wav`. Plot the first 100 samples.
12. Calculate and replay a 2 second chirp. That is, a sinusoid that steadily increases in frequency with time, from say 250Hz at the start to 1000Hz at the end.
13. Build a square wave by adding together 10 odd harmonics: 1f, 3f, 5f, etc. The amplitude of the nth harmonic should be 1/n. Display a graph of one cycle of the result superimposed on the individual harmonics.
14. Write a function called `FtoC` (`ftoc.m`) to convert Fahrenheit temperatures into Celsius. Make sure the program has a title comment and a help page. Test from the command window with:

- i. FtoC(96)
- ii. lookfor Fahrenheit
- iii. help FtoC

15. Write a program to input 2 strings from the user and to print out (i) the concatenation of the two strings with a space between them, (ii) a line of asterisks the same length as the concatenated strings, and (iii) the reversed concatenation. For example:

- i. Enter string 1: Mark
- ii. Enter string 2: Huckvale
- iii. Mark Huckvale
- iv. *****
- v. elavkcuH kraM

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT-I

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

UNIT-II

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

UNIT-III

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

UNIT-IV

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

UNIT-V

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

Suggested Readings

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

Web Sites

1. www.cs.columbia.edu/~nieh/teaching/e6118_s00/
2. www.clarkson.edu/~jnm/cs644
3. pages.cs.wisc.edu/~remzi/Classes/736/Fall2002/

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

UNIT-I

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

UNIT-II

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

UNIT-III

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

UNIT-IV

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

UNIT-V

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

Suggested Readings

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

WEB SITES

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

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

SCOPE

Build and understanding of the fundamental concepts of computer networking. Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

OBJECTIVES

- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry advanced courses in computer networking.
- Understand various transmission media, their comparative study, fiber optics and wireless media.

UNIT I

Introduction: History of Internet – Interconnecting devices – protocols and standards – TCP/IP protocol suite – internetworking devices – routing concept – classfull IP addressing – subnetting – supernetting – classless addressing.

UNIT II

ARP & RARP – Proxy ARP – ARP over ATM – ARP and RAPP protocol format. IP datagram – Fragmentation – Options – IP datagram format – Routing IP datagrams – Cheksum. ICMP – types of messages – message format – Error reporting – Query – Cheksum.

UNIT III

Group management – IGMP message – IGMP operation – process to process communication – UDP operation – TCP services – Flow control.

UNIT IV

BOOTP – DHCP – Address discovery and Binding. DNS – Name Space – DNS in Internet – Resolution – Resource Records.

UNIT V

Remote Login- FTP – SMTP – SNMP . IP over ATM Wan – Cells – Routing the Cells – ATMARF – Logical IP SUBNETS – VPN.%

Suggested Readings

1. Behrouz A. Forouzan (2010). TCP/IP protocol suite (4th ed.). New Delhi: Tata McGraw Hill publication.
2. Andrews S.Tanenbaum (2003). Computer Networks. (4th ed.). New Delhi: Prentice Hall of India Private Ltd.
3. Buck Graham (2007). TCP/IP addressing (2nd ed.). New Delhi: Harcount India Private Limited.
4. Douglas E. Comer (2000). Computer Networks and Internets (4th ed.). New Delhi: Pearson education Asia.
5. William Stallings (2007). Data and Computer Communication Network (8th ed.). New delhi: Tata McGraw Hill

Web sites

1. en.wikipedia.org/wiki/internet_protocol_suite
2. www.yale.edu/pclt/COMM/TCPIP.HTM
3. www.w3schools.com/tcpip/default.asp

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

UNIT – I

Introduction the basics: The Head, the Body, Colors, Attributes, Lists, ordered and unordered.

UNIT – II

Links Introduction: Relative links, absolute links, Link attributes, Using the ID attribute to Link within a Document.

UNIT – III

Images: Putting an image on a page, Using images as Links, Putting an image in the background.

UNIT – IV

Tables: Creating a table, Table headers, Captions, Spanning Multiple columns, styling Table.

UNIT – V

Forms: Basic input and attributes, other kinds of inputs, styling forms with CSS, Where to Go from Here.

Suggested Readings

1. Virginia DeBolt (2006). Integrated HTML and CSS A Smarter, Faster way to learn Wiley/Sybex.
2. Cassidy Williams, Camryn Williams (2015). Introduction to HTML and CSS, O'Reilly.

Websites

1. www.w3schools.com/
2. alecle.net/archives/category/web-technology
3. jmarshall.com/easy
4. www.php.net/
5. en.wikipedia.org/wiki/php
6. www.w3schools.com/PHP/DEfaULT.asp

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT-I

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

UNIT-II

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

UNIT-III

Other XML Concepts: Scripting XML

UNIT-IV

Other XML Concepts: XML as Data, Linking with XML

UNIT-V

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

Suggested Readings

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

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

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

Course Outcomes (COs)

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

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

List of Programs

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

List of Programs

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

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

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

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

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

Course Outcomes (COs)

After successful completion the student will be able to:

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

Implement the following Program

1. Simple router configuration
2. Access and utilize the router to set basic parameters.
3. Connect, configure, and verify operation status of a device interface.
4. Implement static and dynamic addressing services for hosts in a LAN environment.
5. Identify and correct common problems associated with IP addressing and host configurations.
6. Describe basic routing concepts (including: packet forwarding, router lookup process).
7. Configure, verify and troubleshoot RIPv2.
8. Perform and verify routing configuration tasks for a static or default route given
9. Configure, verify and troubleshoot DHCP and DNS operation on a router.
10. Configure and verify a PPP connection between routers.

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

Course Objectives (CO)

- Understand the principles of creating an effective web page, including an in-depth consideration of information architecture.
- Become familiar with graphic design principles that relate to web design and learn how to implement theories into practice.
- Develop skills in analyzing the usability of a web site.
- Understand how to plan and conduct user research related to web usability.
- Learn the language of the web: HTML and CSS.
- Develop basic programming skills using Javascript

Course Outcomes (Cos)

The students will be able to:

1. Analyze a web page and identify its elements and attributes.
2. Design a basic web site using HTML5 and CSS3 to demonstrate responsive web design
3. Implement dynamic web pages with validation using JavaScript objects by applying different event handling mechanism.
4. Create web pages using XHTML and Cascading Style Sheets.
5. Build dynamic web pages using JavaScript (Client side programming).
6. Create XML documents and Schemas

List of Programs

1. Create an HTML document with the following formatting options
 - I. Bold
 - II. Italics
 - III. Underline
 - IV. Headings (Using H1 to H6 heading styles)
 - V. Font (Type, Size and Color)
 - VI. Background (Colored background/Image in background)
 - VII. Paragraph
 - VIII. Line Break
 - IX. Horizontal Rule
 - X. Pre tag
2. Create an HTML document which consists of:
 - I. Ordered List
 - II. Unordered List
 - III. Nested List
 - IV. Image

3. Create an HTML document which implements Internal linkings as well as External linking
4. Create a table using HTML which consists of columns for Roll No., Student's name and grade

Result		
Roll No.	Name	Grade

5. Create a table with the following view

				Place an image here	

6. Create a form using HTML which as the following types of controls:
 - I. Text Box
 - II. Option/radio buttons
 - III. Check boxes
 - IV. Reset and Submit buttons
7. Create HTML documents (having multiple frames) in the following three formats

Frame 1
Frame 2

Frame 1	
Frame2	Frame3

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

Total: 100

Course Objectives (CO)

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

Course Outcomes (COs)

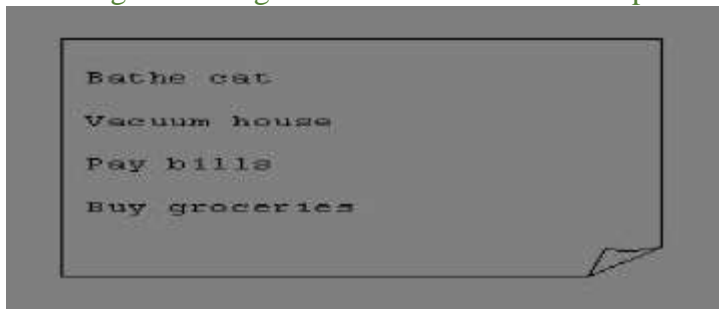
After the completion of this course, a successful student will be able to:

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

1. Information Structure

In this exercise, student will practice identifying the structure of an information object.

For the sample document provided below: Label the information structures you see, including containing structures. 1. Draw a tree representation of the structure.

**2. Deconstructing an XML Document**

In this exercise, student will practice identifying the explicit structure within an XML document. In a sense, this is the reverse of what you did in Exercise #1. For the sample

XML markup below, create a document-like representation (or a simple drawing) for the content contained within the XML tags:

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

3. Creating XML Markup

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

4. Well-Formedness

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

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

5. Well Formedness

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

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

16CTU501A CRYPTOGRAPHY AND NETWORK SECURITY 4H – 4C**Instruction Hours / week: L: 4 T: 0 P: 0 Marks: Int : 40 Ext : 60 Total: 100****Course Objectives (CO)**

- To understand basics of Cryptography and Network Security.
- To know about various encryption techniques.
- To understand the concept of Public key cryptography.
- To study about message authentication and hash functions
- To impart knowledge on Network security
- To understand various protocols for network security to protect against the threats in the networks.

Course Outcomes (COs)

After successful completion of the course, the learners would be able to:

1. Illustrate various Public key cryptographic techniques
2. Evaluate the authentication and hash algorithms
3. Provide security of the data over the network.
4. Do research in the emerging areas of cryptography and network security.
5. Implement various networking protocols.
6. Protect any network from the threats in the world

UNIT-I

Introduction to Cryptography – security attacks – Security services- security Algorithm – Stream cipher and Block cipher – Symmetric and Asymmetric – key cryptosystems; Symmetric key algorithms: Introduction – DES – Triple DES – AES – IDEA – Blowfish – RC5.

UNIT-II

Public key Cryptosystem: Introduction to Number Theory – RSA Algorithm – key management – Diffie-Hellman key exchange – Introduction to elliptic curve cryptography; message authentication and hash function – hash and MAC algorithm – digital signatures and authentication protocol.

UNIT-III

Network security practice: Authentication applications – kerberos – X.509 authentication services and encryption techniques;;E-mail security – PGP – S/MIME – IP security

UNIT-IV

Web security – secure socket layer – secure electronic transaction; System security – Intruders and viruses – Firewalls – Password security.

UNIT V

Case study : Implementation of Cryptographic algorithms – RSA – DSA – ECC (C/JAVA programming). Network forensic -Security audit; Other security mechanism:

Introduction to steganography – quantum cryptography – Water marking – DNA cryptography.

Suggested Readings

1. William Stallings (2013). Cryptography and Network Security (6th ed.). Pearson education, New delhi.
2. Bruce Schneir (2016). Applied cryptography (2nd ed.). CRC Press, New delhi.
3. A.Menezes, P.Van Oorschot and S.Vanstone (2010). Handbook of applied cryptography (2nd ed.). CRC Press New delhi.
4. Ankit Fadia (2010). Network security (2nd ed.). McMillan India Ltd., New Delhi.

Websites

1. williamstallings.com/crypto3e.html
2. u.cs.biu.ac.il/~herzbea/book.html
3. www.flipkart.com/search-books/cryptography+and+network+security+William+stallings+ebook

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT – I Testing Fundamentals

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

UNIT – II Examining the code

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

Flowgraphs and Path Testing

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

UNIT – III Transaction-Flow Testing and Data-Flow Testing

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

UNIT – IV Domain Testing

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

UNIT – V Logic-Based Testing and State Graphs

Motivational Overview-Decision Tables-Path Expressions Again-KV Charts-Specifications

State Graphs-Good State Graphs and Bad-State Testing

Suggested Readings

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

Web Sites

1. www.testinggeek.com
2. www.softwaretestinghelp.com
3. www.softwaretestinginstitute.com

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

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

Course Outcomes (COs)

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

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

UNIT I

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

UNIT II

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

UNIT III

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

UNIT IV

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

UNIT V

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

Suggested Readings

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

Websites

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT-I

Transport Layer Protocols: TCP, UDP, SCTP protocol.

UNIT-II

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

UNIT-III

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

UNIT-IV

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

UNIT-V

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

Suggested Readings

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

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

- To practice statistical and programming best practices
- To develop professional skills: clear communication, creative thinking, critical thinking, self-directed learning, and effective teamwork
- To understand data analysis techniques and an interest in improving their ability to tackle data-rich problems in a systematic, principled way.
- To understand the basic concepts of big data, methodologies for analyzing structured and unstructured data with emphasis on the relationship between the Data Scientist and the business needs.
- To develop the skills required by organizations to support these functions has been grouped under the term “Data Sciences”

Course Outcomes (COs)

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

1. Understand the problems solvable with data science and an ability to attack them from a statistical perspective.
2. Understand when to use supervised and unsupervised statistical learning methods on labeled and unlabeled data-rich problems.
3. Create data analytical pipelines and applications in Python.
4. Familiar data science ecosystem and the various tools needed to continue developing as a data scientist.
5. Develop the mindset to work like a data scientist, and follow a methodology to tackle different types of data science problems
6. Describe what data science and machine learning are, their applications & use cases, and various types of tasks performed by data scientists

UNIT-I

Data Scientist’s Tool box: Turning data into actionable knowledge, introduction to the tools that will be used in building data analysis software; version control, markdown, git, GitHub, R and R-Studio.

UNIT-II

R Programming basics: Overview of R, R datatypes and Objects, reading and writing data, control structures, functions, scoping rules, dates and times, loop functions, debugging tools, simulation, code profiling.

UNIT-III

Getting and cleaning data: Obtaining data from the web, from API’s, from database, and from colleagues in various formats. Basics of data cleaning and making data –tidy.

UNIT-IV

Exploratory data analysis: Essential exploratory techniques for summarizing data, applied before formal modelling commences, eliminating or sharpening potential hypotheses about the world that can be addressed by the data, common multivariate statistical techniques used to visualize high-dimensional data.

UNIT-V

Reproducible Research: Concepts and tools behind reporting modern data analysis in a reproducible manner, To write a document using R markdown, integrate live R code into a literate statistical program, compile R markdown documents using knitr and related tools, and organize a data analysis so that it is reproducible and accessible to others.

Suggested Readings

1. Rachel Schutt, Cathy O’Neil (2013). Doing data science: Straight Talk from the frontline. S.Chroff/O’ Reilly
2. Foster Provost, Tom Fawcett (2013). Data science for Business – What you need to know about Datamining and Data Analytic Thinking. O’Reilly
3. John. W. Foreman (2013). Data Smart: Using Data science to transform information into insight. John Wiley and Sons.
4. Ian Ayres (2007). Super Crunchers: Why Thinking – By – Numbers is the New way to Be Smart (1st ed.). Bantam
5. Eric Seigel (2013). Predictive Analytics: The Power of Predict who will click, Buy Lie or Die (1st ed.). Wiley.
6. Matthew A.Russel (2013). Mining the social webL: Datamining Facebook, Twitter, LinkedIn, Google+, GitHub, and More (2nd ed.). O’Reilly Media

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT-I

Overview: Predictive and descriptive data mining techniques

UNIT-II

Supervised and unsupervised learning techniques

UNIT-III

Process of knowledge discovery in databases, pre-processing methods

UNIT-IV

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

UNIT-V

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

Suggested Readings

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

WEB SITES

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

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

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

Course Outcomes (COs)

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

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

UNIT-I

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

UNIT-II

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

UNIT-III:

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

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

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

Course Outcomes (COs)

Upon successful completion the student will be able to:

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

UNIT I

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

UNIT II

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

UNIT III

Introducing photoshop elements: About elements – welcome screen – create mode – menu bar – toolbox – options bar – panels. Organizing images: Obtaining images -

tagging images – searching for images – opening and saving images. Selecting areas – Layers – Text and drawing tools.

UNIT IV

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

UNIT V

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

Suggested Readings

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

Websites

1. en.wikipedia.org/wiki/Multimedia
2. www.arena-multimedia.com/
3. www.nextwavemultimedia.com/

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

Total: 100

Course Objectives (CO)

- To understand basics of Cryptography and Network Security.
- To know about various encryption techniques.
- To understand the concept of Public key cryptography.
- To study about message authentication and hash functions
- To impart knowledge on Network security
- To understand various protocols for network security to protect against the threats in the networks.

Course Outcomes (COs)

After successful completion of the course, the learners would be able to:

1. Illustrate various Public key cryptographic techniques
2. Evaluate the authentication and hash algorithms
3. Provide security of the data over the network.
4. Do research in the emerging areas of cryptography and network security.
5. Implement various networking protocols.
6. Protect any network from the threats in the world

List of Programs

1. Write a program to convert your college name from plain text to cipher text using transposition cipher method of encryption
2. Write a program to convert your name from plain text to cipher text using the One Time Pads method of encryption
3. Write a program to encrypt a paragraph using the data encryption standard algorithm
4. Write a program to encrypt your bio-data using the advanced encryption standard algorithm
5. Write a program to decrypt the “Network Security” theory syllabus using the RSA Algorithm.
6. Write a program that takes a binary file as input and performs bit stuffing and cyclic redundancy check computation.
7. Write a program to simulate the working of Sliding-window protocol
8. Write a program to find the shortest path in a network using Dijkstra’s Algorithm
9. Write a program to implement the token bucket algorithm for congestion control.
10. Write a program for the following chat applications:

One to One: Open a Socket connection and display what is written by one to another.

Many to Many: Each client opens a socket connection to the client server and writes to the Socket. Whatever is written by one can be seen by all. Implement symmetric key cryptography.

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

List of Programs

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

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

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

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

Course Outcomes (COs)

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

1. Grasp the fundamentals of a programming language and know the basic differences between programming languages
 2. Choose the architecture based on the problem to be solved.
 3. Differentiate between the types of applications supported by .Net
 4. Build, compile and execute a VB .Net program
 5. Apply techniques to develop error-free software
 6. To build integrated VB.NET solutions using files and structures with printing capabilities. Translate general requirements into data-related solutions using database concepts
-
1. Write a VB .Net program to calculate simple interest and compound interest.
 2. Write a VB .Net program to implement Calculator.
 3. Write a VB.Net program to implement Notepad
 4. Write a VB.Net program to draw several shaps and fill with color.
 5. Write a VB .Net program to perform the following in list box
 - a) Add an item
 - b) Delete an item
 - c) List count
 - d) Clear the list
 6. Write a program to calculate the total marks of the student and print the grades.
 7. Write a VB .Net program to implement employee payroll system
 8. Write a VB .Net program to create and manipulate a file.
 9. Write a program to implement a web browser

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

16CTU513A INTRODUCTION TO DATA SCIENCE – PRACTICAL 4H – 2C

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

- To practice statistical and programming best practices
- To develop professional skills: clear communication, creative thinking, critical thinking, self-directed learning, and effective teamwork
- To understand data analysis techniques and an interest in improving their ability to tackle data-rich problems in a systematic, principled way.
- To understand the basic concepts of big data, methodologies for analyzing structured and unstructured data with emphasis on the relationship between the Data Scientist and the business needs.
- To develop the skills required by organizations to support these functions has been grouped under the term “Data Sciences”

Course Outcomes (COs)

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

7. Understand the problems solvable with data science and an ability to attack them from a statistical perspective.
8. Understand when to use supervised and unsupervised statistical learning methods on labeled and unlabeled data-rich problems.
9. Create data analytical pipelines and applications in Python.
10. Familiar data science ecosystem and the various tools needed to continue developing as a data scientist.
11. Develop the mindset to work like a data scientist, and follow a methodology to tackle different types of data science problems
12. Describe what data science and machine learning are, their applications & use cases, and various types of tasks performed by data scientists

List of Progrms

1. Write a program that prints ‘Hello World’ to the screen
2. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n.
3. Write a program that prints multiplication table for numbers upto 12.
4. Write a function that returns the largest element in a list
5. Write a function that computes the running total of a list
6. Write a function that tests whether string is a palindrome.
7. Implement linear search
8. Implement binary search
9. Impelment matrix addition, subtraction and multiplication
10. Fifteen students were enrolled in a course
 There ages were : 20 20 20 20 20 21 21 21
 22 22 22 22 23 23 23
 i. Find the median age of all students under 22 years.

- ii. find the median age of all students
- iii. Find the mean age of all students
- iv. Find the modal age of all students
- v. Two more students enter the class. The age of both students is 23. What is now mean, mode and median.

11. Following table gives frequency distribution of systolic blood pressure. Compute all the measures of dispersion.

Midpoint	95.5	105.5	115.5	125.5	135.5	145.5	155.5	165.5	175.5
Number	5	8	22	27	17	9	5	5	2

- 12. Obtain probability distribution of, where X is number of spots showing when a six-sided symmetric die (i.e. all six faces of the die are equally likely) is rolled. Simulate random samples of sizes 40, 70 and 100 respectively and verify the frequency interpretation of probability.
- 13. Make visual representations of data using the base, lattice and ggplot2 plotting systems in R, apply basic principles of data graphics to create rich analytic graphics from available datasets.
- 14. Use Git/GitHub software to create Github Account. Also, create a repo using Github.

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

List of Programs

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

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

1. Review the fundamental concepts of a digital image processing system.
2. Analyze images in the frequency domain using various transforms.
3. Evaluate the techniques for image enhancement and image restoration.
4. Categorize various compression techniques.
5. Interpret Image compression standards.
6. Interpret image segmentation and representation techniques.
- 7.

List of Programs

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

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

Upon successful completion the student will be able to:

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

List of Programs

1. Change a Shape to Another Shape. (Shape Animation)
2. Create a Man to walk with the help of Key Frame Animation.
3. Change the Colors of an object with the help of Animation.
4. Draw a Bird with Flash tools and make it fly with key Frame Animation.
5. Create a Shining Stones with the help of Movie Clip.
6. Create an animation to represent a growing moon using frame by frame animation
7. Create an animation to bounce a ball on steps.
8. Simulate movement of a cloud.
9. Create Morphing between two images.
10. Create an Action script to execute for an event in an application.
11. Create Water Drops using Photoshop.

12. Animate Plane Flying with the Clouds.
13. Create Plastic Surgery for Nose.
14. Create a Web Page using Photoshop
15. Given a picture of a flower with a background. Extract the flower and organize on a different background.
16. Display the given picture through your name using mask.

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT-I**Introduction to PHP:**

PHP introduction, inventions and versions, important tools and software requirements (like Web Server, Database, Editors etc.) -PHP with other technologies, 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

UNIT-IV

String Manipulation and Regular Expression: (3L)

- Creating and accessing String , Searching & Replacing String
- Formatting, joining and splitting String , String Related Library functions
- Use and advantage of regular expression over inbuilt function
- Use of preg_match(), preg_replace(), preg_split() functions in regular expression.

UNIT-V

Array:

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

Suggested Readings

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT-I

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT I -An Introduction to Electronic commerce

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

UNIT II -The Internet and WWW

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

UNIT III: Electronic data

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

UNIT IV: Planning for Electronic Commerce

Planning Electronic commerce initiatives, linking objectives to business strategies, measuring cost objectives, comparing benefits to Costs, strategies for developing electronic commerce web sites.

UNIT V : Internet marketing

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

Suggested Readings

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

Websites

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

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

UNIT-I

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

UNIT-II

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

UNIT-III

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

UNIT-IV

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

UNIT-V

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

Suggested Readings

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

WEB SITES

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

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

- To familiarize with soft computing concepts.
- To introduce the fuzzy logic concepts, fuzzy principles and relations.
- To Basics of ANN and Learning Algorithms.
- Develop the skills to gain a basic understanding of neural network theory and fuzzy logic theory.
- Genetic Algorithm and its applications to soft computing.
- Hybrid system usage, application and optimization

Course Outcomes (COs)

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

1. List the facts and outline the different process carried out in fuzzy logic, ANN and Genetic Algorithms.
2. Explain the concepts and meta-cognitive of soft computing.
3. Apply Soft computing techniques the solve character recognition, pattern classification, regression and similar problems.
4. Outline facts to identify process/procedures to handle real world problems using soft computing.
5. Evaluate various techniques of soft computing to defend the best working solutions.
6. Design hybrid system to revise the principles of soft computing in various applications.

UNIT I**Neural Networks – 1 (Introduction a Architecture)**

Neuron, Nerve structure and synapse, Artificial Neuron and its model, activation functions, Neural network architecture: single layer and multilayer feed forward networks, recurrent networks. Various learning techniques; perception and convergence rule, Auto associative and hetro-associative memory.

UNIT II

Neural Networks – II (Back propagation networks) architecture: perceptron model, solution, single layer artificial neural network, multilayer perceptron model, back propagation learning methods, effect of learning rule co-efficient; back propagation algorithm, factors affecting back propagation training, applications.

UNIT III**Fuzzy Logic – 1 (Introduction)**

Basic concepts of fuzzy logic, Fuzzy sets and Crisp sets, Fuzzy set theory and operations, Properties of fuzzy sets, Fuzzy and Crisp realtions, Fuzzy to Crisp conversion

UNIT IV

Fuzzy Logic II (Fuzzy membership, Rules) Membership functions, interference in fuzzy logic, fuzzy if-then rules, Fuzzy implications and Fuzzy algorithms, Fuzzyfications & Defuzzifications, Fuzzy Controller, Industrial applications.

UNIT V

Genetic Algorithm (GA)

Basic concepts, working principle, procedures of GA, flowchart of GA, Genetic representations, (encoding) Initialization and selection, genetic operators, Mutation, Generational Cycle, applications

Suggested Readings

1. S.Rajasekaran & G.A. Vijayalakshmi Pai. Neural Networks, Fuzzy Logic and Genetic Algorithm: Synthesis and Applications. Prentice Hall of India.
2. N.P.Padhy. Artificial Intelligence and Intelligent Systems. Oxford University Press.
3. Siman Haykin. Neural Networks. Prentice Hall of India
4. Timothy J.Ross. Fuzzy Logic with Engineering Applications. Wiley India.
5. Kumar Satish. Neural Networks. Tata McGraw Hill.

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

Course Objectives (CO)

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

Course Outcomes (COs)

Upon completion of the subject, students will be able to

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

UNIT-I

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

UNIT-II**Lexical Analysis:**

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

UNIT-III**Parsing:**

Bottom up parsing- LR parser, yaITU. **Intermediate representations: Three** address

code generation, syntax directed translation, translation of types, control Statements.

UNIT-IV

Storage organization: Activation records stack allocation.

UNIT-V

Code Generation: Object code generation

Suggested Readings

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

Course Objectives (CO)

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

Course Outcomes (COs)

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

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

List of Programs

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

8. Write a PHP script that removes the whitespaces from a string.
Sample string : 'The quick brown fox' Expected Output : Thequickbrownfox
9. Write a PHP script that finds out the sum of first n odd numbers.
10. Create a login page having user name and password. On clicking submit, a welcome message should be displayed if the user is already registered (i.e.name is present in the database) otherwise error message should be displayed.
11. Write a PHP script that checks if a string contains another string.
12. Create a simple 'birthday countdown' script, the script will count the number of days between current day and birth day.
13. Create a script to construct the following pattern, using nested for loop.
- ```
*
* *
* * *
* * * *
* * * * *
```
14. Write a simple PHP program to check that emails are valid.
15. WAP to print first n even numbers.
16. \$color = array('white', 'green', 'red')
- Write a PHP script which will display the colors in the following way : Output :  
white, green, red,  
• green • red  
• white
17. Using switch case and dropdown list display a —Hello! message depending on the language selected in drop down list.
18. Write a PHP program to print Fibonacci series using recursion.
19. Write a PHP script to replace the first 'the' of the following string with 'That'.
- Sample :** 'the quick brown fox jumps over the lazy dog.'  
**Expected Result :** That quick brown fox jumps over the lazy dog.

**16CTU611B UNIX / LINUX PROGRAMMING - PRACTICAL 4H – 2C****Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100****Course Objectives (CO)**

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

**Course Outcomes (COs)**

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

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

**List of Programs**

1. Write a shell script to check if the number entered at the command line is prime or not.
2. Write a shell script to modify —call command to display calendars of the specified months.
3. Write a shell script to modify —call command to display calendars of the specified range of months.
4. Write a shell script to a)T)Ue)pt 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

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

**Course Objectives (CO)**

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

**Course Outcomes (COs)**

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

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

**List of Programs**

1. Write a HTML program to implement the use of Image map.
2. Write a CSS to implement selectors in HTML
3. Write a CSS to implement pseudo – classes with in-line styles
4. Write a Javascript program to validate a web form
5. Write a Javascript program to allow visitors to see history of visiting your page
6. Write a Javascript program to change random color each 5 seconds

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

**Course Objectives (CO)**

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

**Course Outcomes (COs)**

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

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

**List of Programs**

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

**Course Objectives (CO)**

- To familiarize with soft computing concepts.
- To introduce the fuzzy logic concepts, fuzzy principles and relations.
- To Basics of ANN and Learning Algorithms.
- Develop the skills to gain a basic understanding of neural network theory and fuzzy logic theory.
- Genetic Algorithm and its applications to soft computing.
- Hybrid system usage, application and optimization

**Course Outcomes (COs)**

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

1. List the facts and outline the different process carried out in fuzzy logic, ANN and Genetic Algorithms.
2. Explain the concepts and meta-cognitive of soft computing.
3. Apply Soft computing techniques the solve character recognition, pattern classification, regression and similar problems.
4. Outline facts to identify process/procedures to handle real world problems using soft computing.
5. Evaluate various techniques of soft computing to defend the best working solutions.
6. Design hybrid system to revise the principles of soft computing in various applications.

**List of Programs**

1. Implement OR, AND using Perceptron in C
2. Implement OR,AND using Perceptron in MATLAB Command-line Argument
3. Implement OR,AND using Perceptron in MATLAB GUI
4. Implement OR,AND, X-OR gate, using back propagation algorithm in MATLAB using Command-line Argument as well as GUI
5. Solve a given problem – 1 (Operatons) using Fuzzy Ligic in MATLAB
6. Solve a given problem – 1 (Max-Min Composition) using Fuzzy Ligic in MATLAB
7. To find the solution of the function Maximize, given the constraints using GA approach in C.
8. Solve a given problem – 1 using Fuzzy Logic in MATLAB GUI
9. Study GA tool in MATLAB.

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

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

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

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

Upon completion of the subject, students will be able to

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

### **List of Programs**

1. 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 yacc: write a desk calculator.