BACHELOR OF COMPUTER APPLICATIONS (BCA)

CHOICE BASED CREDIT SYSTEM (CBCS)

Syllabus (2017 – 2018)

DEPARTMENT OF COMPUTER SCIENCE, APPLICATIONS & INFORMATION TECHNOLOGY
FACULTY OF ARTS, SCIENCE AND HUMANITIES

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC Act, 1956)
Eachanari (Post), Coimbatore – 641 021.
Phone No. 0422-6471114, 6471115, 6453777
Fax No: 0422-2980022-3
E mail ID: info@karpagam.com
Web: www.kahedu.edu.in
Semester – I

பகுதி – I, தமிழ்
17LSU101: தமிழ் முதல் தாள் 4-H,4-C
(தொடரில் ஆங்கிலம் பாடல் வாங்கப்படுகிறது)

அலகு - I: சிக்கச்சொல்லியம்:
(10 மணிநேரம்)

அலகு - II: ஆயத்தியம்:
(10 மணிநேரம்)

அலகு - III: இருவியக்கியம்:
(10 மணிநேரம்)

அலகு - IV:
(10 மணிநேரம்)

1. மாநிலச் சிவிலியன் - உலகியலைக்குறிக்கான
2. மாநிலச் சிவிலியன் - அ. சூரியச் சிவிலியன்
3. மாநிலச் சிவிலியன் - குறியீடுகள்
4. உலகியலைக்குறிக்கான ஆங்கிலம் - யாராண்டை
5. எல்லாச் சிவிலியன் - இடையக்கருவம்

அலகு - V: பாடல் பாடல்
(8 மணிநேரம்)

1. தமிழ் பாடல் முதல் பாடல் பாடல் முறை
2. பாடல் பாடல் முறை
3. பாடல் பாடல் முறை
4. பாடல் பாடல் முறை

பாடல்: கூறுகற்றூர் - குறித் தோற். கூறுகற்றூர் வாணங்கலரக்குறிக்கான குறித் தோற் பாடல்
Scope
This course provides student with a comprehensive study of the fundamentals of C and C++ programming language. Classroom lectures provide students the efficient code writing skills.

Objectives
- Write C programs that are non-trivial.
- Utilize the modular features of the language.
- Utilize pointers to efficiently solve problems.
- Use functions from the portable C library.
- To familiarize the students with the concepts of computer programming.
- To present the syntax and semantics of the “C++” language as well as basic data types offered by the language.
- To discuss the principles of the object-oriented model and its implementation in the “C++” language.
- To demonstrate the means useful in resolving typical implementation problems with the help of standard “C++” language libraries.

Unit-I

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

Expressions, Conditional Statements and Iterative Statements: Simple Expressions in C++ (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch-case construct), Understanding syntax and utility of Iterative Statements (while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)
Unit-II

Functions and Arrays: Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments. Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays.

Unit-III

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

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

Unit-IV

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

File I/O, Preprocessor Directives: Opening and closing a file (use of fstream header file, ifstream, ofstream and fstream classes), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros.
Unit-V
Using Classes in C++: Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Function overloading in classes, Class Variables & Functions, Objects as parameters, Specifying the Protected and Private access, Copy Constructors, Overview of Template classes and their use.
Overview of Function Overloading and Operator Overloading: Need of Overloading functions and operators, Overloading functions by number and type of arguments, Looking at an operator as a function call, Overloading Operators (including assignment operators, unary operators)
Inheritance, Polymorphism and Exception Handling: Introduction to Inheritance (Multi-Level Inheritance, Multiple Inheritance), Polymorphism (Virtual Functions, Pure Virtual Functions), Basics Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions, Restricting exceptions, Rethrowing exceptions.

Suggested Readings
Websites

2. http://www2.its.strath.ac.uk/courses/c/
5. www.cplusplus.com/
Scope
Computer System Architecture deals with the architecture of computer systems with its various processing units and also the performance measurement of the computer system. This course is designed to provide a comprehensive introduction to digital logic design leading to the ability to understand number system representations, binary codes, binary arithmetic and Boolean algebra, and its relevance to digital logic design.

Objectives
- 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 Cache memory and its importance

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
Suggested readings
Scope
This course gives students a brief knowledge on ways of storing, accessing, processing and transmitting information. It also provides elements of computer system, data, CPU, memory and internet.

Objectives
- Understand the concepts of Information Technology and its applications.
- Becomes familiar with basic knowledge of Information Technology.

Unit- I

Unit-II
Acquiring graphical data-Introduction-Acquisition of textual data and pictures-Storage format for pictures-Image compression fundamentals-Image acquisition with digital camera-Acquiring audio data-Acquisition of video processing textual and multimedia data.

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

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

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

17CAU103 INTRODUCTION TO INFORMATION TECHNOLOGY  4H – 4C

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

Suggested readings

Websites
1. www.tesu.edu/watson/IT
2. www.bcanotes.com/It.html
3. www.amazon.in/Fundamentals of IT
1. Write a program to print the sum and product of digits of an integer.

2. Write a program to reverse a number.

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

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

5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.

6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.

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

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

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

   *
   ***
   *****
   *******
   **********

10. Write a program to perform following actions on an array entered by the user:

    i) Print the even-valued elements
    ii) Print the odd-valued elements
    iii) Calculate and print the sum and average of the elements of array
    iv) Print the maximum and minimum element of array
    v) Remove the duplicates from the array
    vi) Print the array in reverse order
11. Write a program that swaps two numbers using pointers.

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

13. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.

14. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.

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

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

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

(Any 8 Experiments)

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
1. Create a set of slides with sound and animation using power point and to select various styles of slides from slide template.

2. Prepare an organization chart for a company and a college using power point.

3. To perform comparative study of your UG degree subject marks and create a various styles of graph using excel.

4. To perform student and hostel fee particulars in a worksheet. Analysis the results using excel.

5. To create a word document for salary preparation and access the calculation, graph from excel using OLE.

6. Apply the mail merge concept using word.


8. Animation using Java.

9. Create simple accounts using Tally.

10. Create balance sheet using Tally.
Scope
The study creates awareness among the people to know about various renewable and nonrenewable resources of the region, enables environmentally literate citizens (by knowing the environmental acts, rights, rules, legislation, etc.) to make appropriate judgments and decisions for the protection and improvement of the earth.

Objectives
- Creating the awareness about environmental problems among people.
- Developing an attitude of concern for the environment.
- Motivating public to participate in environment protection and improvement.

Unit-I

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

Unit-III
Unit- IV


Unit-V


Suggested Readings
பகுதி – I, தமிழ்

17LSU201: நடுவணவம் தாள் 4-H,4-C
(தமிழில் அருங்கையம் பட்டு முதியாதைப்புறத்து)

அகத்தியம் - I: பகுதி விளக்கப்பட்டினம்

1. காலாதை - மோசாபோரம் - குழுக்குடைவாசம் பாராமை.

2. கலைசம் - மோசாபோரம் குழுக்குடைவாசம்: 10 பாராமை.

அகத்தியம் - II: கலைசம் விளக்கப்பட்டினம்

(15 மல்லிகராம்)

அகத்தியம் - III: காப்பியம் (6 மல்லிகராம்)

பத்துப்பாட்டு - (21-29) - கலைசம் விளக்கப்பட்டினம்.
அலகு IV: சிறுகறத (10 மாண்டுப் பயிற்சி)

1. கருடாமயன் ஆசைவன் - ம.சி.குலான்
2. கருடாமயன் குறுக்கு - பியோல
3. கருடாமயன் – கோழிக்கோட்டையார்
4. கருடாமயன் – கோவை

அலகு V: பமாழிப்பயிற்சி (7 மாண்டுப் பயிற்சி)

பமாழிப்பிள்ளை பயிற்சிகள் (கல்வி, கல்வி, கல்வி, கல்வி)

பமாழிப்பம்: 

பாட்டுகள்: கல்வி, கல்வி, கல்வி, கல்வி, 

பமாழிப்பம்: கல்வி, கல்வி.
Scope
This course offers an introduction to the Java programming language. This course covers the basic topics considered are programs and program structure in general, and Java syntax, data types, flow of control, classes, methods, objects, arrays, exception handling, recursion, and graphical user interfaces (GUIs).

Objectives
- Understand the basic concepts and principles of structured programming.
- Understand the basic concepts and principles of object oriented programming.
- basic java programming and Applet programming
- understand the fundamental of Packages and access modifiers and interface in java
- understand the fundamental of Exception Handling and AWT component and AWT classes

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, text fields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts. Overview of servlets.

**Suggested readings:**


**Websites**

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

Scope
It exposes the students to study entities such as sets, relations, graphs, and trees. These entities act as very fundamental representations useful in a broad spectrum of applications across the length and breadth of computer science.

Objectives
This course provides a deep knowledge to the learners to develop and analyze algorithms as well as enable them to think about and solve problems in new ways. By the completion of the course students should be able to express ideas using mathematical notation and solve problems using the 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
Prepositional Logic: Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory.
Suggested readings


Scope
This course provides an introduction to the script programming paradigm, and introduces scripting languages HTML, DHTML, XML and PHP.

Objectives:
To acquaint students with
- a variety of scripting languages, including DHTML, PHP, Javascript, and XML.
- the skill to create and format dynamic web pages, the collecting information from web pages.
- The skills to write server side scripts and design web applications and implementing applications using appropriate script language.

Unit- I

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

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

Unit-IV

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

Suggested readings

Websites
1. http://www.mvps.org/scripting/languages
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 aCSUess specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
13. Write a program to show the use of static functions and to pass variable length arguments in a function.

14. Write a program to demonstrate the concept of boxing and unboxing.

15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).

16. Write a program to create a multilevel package and also creates a reusable class to generate FibonaCSUi series, where the function to generate 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 —DivideByZerol that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.

19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.

20. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).

21. Write a program to demonstrate priorities among multiple threads.

22. Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).

23. Write a program to create URL object, create a URLConnection using the openConnection() method and then use it examine the different components of the 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 scrolls the message in the banner from left to right across the applet's window.

26. Write a program to get the URL/location of code (i.e. java code) and document(i.e. html file).

27. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().

28. Write a program to demonstrate different keyboard handling events.

29. Write a program to generate a window without an applet window using main() function.

30. Write a program to demonstrate the use of push buttons.
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
   \[ \frac{1}{1\times 2} + \frac{1}{2\times 3} + \ldots + \frac{1}{n(n+1)} = \frac{n}{n+1} \]
4. Write a C Program to perform the sum \(= 1 + (1+2) + (1+2+3) + \ldots + (1+2+\ldots+n)\)
5. Write a C program to print Fibonacci series till Nth term using recursion
6. Write a C program in c to calculate factorial of a number using recursion
7. Write a C Program to find a minimum spanning tree using Prim’s algorithm
8. Write a C program to find the shortest path with the lower cost in a graph using Dijkstra’s Algorithm
9. Write a C Program to construct the truth table for the following formula.
   (i) \(P \wedge Q \wedge R\)    (ii) \(P \lor Q \wedge R\)    (iii) \(P \wedge Q \wedge R\)
10. Write a C Program to prove De – Morgan’s law.
1. Create a form to reserve a ticket in the railways if the source and destination place is given.
2. Create a web page to display student mark statement.
3. Design an home page for a company
4. Develop DHTML Script to illustrate Color and Background attribute.
5. Create a DHTML page using various filters on images, mask image, mask Text.
6. Design a PHP program to find greatest of three numbers.
7. Design an student application form using PHP program.
8. Write a java script program design a calculator
9. Develop a Java script program for display greetings based on Time
10. Create an XML document for student information with relevant attributes and validation.
11. Develop a program to copy the content of one file to another file using PHP program.
12. Develop an E-mail application using PHP program.
Instruction Hours / week: L: 4 T: 0 P: 0  Marks: Internal: 100  Total: 100

Objectives: To train students in acquiring proficiency in English by reading different genres in literature and learning grammar
To provide aesthetic pleasure through literature

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

UNIT - II : POEM
1. The Stolen Boat - William Wordsworth
3. The Sailor - Safaa Fathy

UNIT - III : SHORT STORIES
1. Rapunzel - Brothers Grimm
2. The Romance of a Busy Broker - O.Henry
3. The Nightingale and the Rose - Oscar Wilde.

UNIT - IV
GRAMMAR : 1. Tenses
2. Auxiliaries (Primary and Modal)
3. Articles
4. Tag Questions

UNIT - V

FUNCTIONAL ENGLISH
1. Filling the blanks with the suitable form of verb in a conditional sentence.
2. Dialogue Writing
3. Changing positive to negative without altering the meaning
4. Fill in the blank with suitable modal
5. Framing a question to a statement
6. Rewrite the sentences changing the underlined word as directed

Prescribed Text: Reminisce, Published by the Department of English, Karpagam University.

Semester – III

17CAU301 DATA STRUCTURES 4H – 4C

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

Scope
Data structures and algorithms are the building blocks in computer programming. This course will give students a comprehensive introduction of common data structures, and algorithm design and analysis. This course also intends to teach data structures and algorithms for solving real problems that arise frequently in computer applications, and to teach principles and techniques of computational complexity.

Objectives
- understand common data structures and algorithms, and be able to implement them
- Judge efficiency trade-offs among alternative data structure implementations or combinations.
- analyze the complexities of data structures and algorithms;
- choose appropriate data structures and algorithms for problem solving.

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

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

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

Unit-IV
Searching and Sorting, Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Insertion Sort, Shell Sort, Comparison of Sorting Techniques
Unit-V
Hashing - Introduction to Hashing, Deleting from Hash Table, Efficiency of Rehash Methods, Hash Table Reordering, Resolving collusion by Open Addressing, Coalesced Hashing, Separate Chaining, Dynamic and Extendible Hashing, Choosing a Hash Function, Perfect Hashing, Function

Suggested readings

Websites
Scope
This course introduces the concepts and principles of operating systems, provide students with the basic knowledge and skills of memory, device and Process management and techniques.

Objectives
- Understand the principles of concurrency and synchronization.
- Understand basic resource management techniques.
- Understand the implementation of fundamental OS structures, including Threads, processes, system calls, scheduling, virtual memory, and file systems

Unit-I

Unit-II

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
Suggested readings


Websites

1. www.cs.columbia.edu/~nieh/teaching/e6118_s00/

2. www.clarkson.edu/~jnm/cs644/pages.cs.wisc.edu/~remzi/Classes/736/Fall2002/
Scope
The course includes basics of switched communication networks, TCP/IP networking, network programming, packet switch architecture, rate and congestion control, Quality-of-Service networks, wireless communications.

Objectives
- Various transmission media, their comparative study, fiber optics and wireless media
- Categories and topologies of networks (LAN and WAN) Layered architecture (OSI and TCP/IP) and protocol suites.
- Channel error detection and correction, MAC protocols, Ethernet and WLAN.
- Details of IP operations in the INTERNET and associated routing principles

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; packetswitching - connectionless datagram switching, connection-oriented virtual circuit switching; dial-up modems; digital subscriber line; cable TV for data transfer.

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

**Multiple Access Protocol and Networks:** CSMA/CD protocols; Ethernet LANS; connecting LAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways;

**Networks Layer Functions and Protocols:** Routing; routing algorithms; network layer protocol of Internet- IP protocol, Internet control protocols.

Unit-V

**Transport Layer Functions and Protocols:** Transport services- error and flow control, Connection establishment and release- three way handshake;

**Overview of Application layer protocol:** Overview of DNS protocol; overview of WWW & HTTP protocol.

**Suggested readings**


**Websites**

1. en.wikipedia.org/wiki/Internet_protocol_suite
3. www.yale.edu/pclt/COMM/TCPIP.HTM
4. www.w3schools.com/tcpip/default.asp
5. https://www.cse.iitb.ac.in/
Scope
This is an introductory Android programming course designed to introduce and familiarize participants with programming in the Android environment.

Objectives
- to explain the differences between Android and other mobile development environments.
- teach students to design, create, deploy, and test applications for the Android mobile phone platform.
- introduce students to the most common tools and techniques for writing Android applications.
- to explain how Android applications work, their life cycle, manifest, Intents, and using external resources.
- to teach to access and work with databases under android OS
- design and develop useful Android applications with compelling user interfaces .by using, extending, and creating your own layouts and Views and using Menus.

Unit-I

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

Unit-III
Development Tools: Installing and using Eclipse with ADT plug-in, Installing Virtual machine for Android sandwich/Jelly bean (Emulator), configuring the installed tools, creating a android project– Hello Word, run on emulator, Deploy it on USB-connected Android device. (5L)

Unit-IV
User Interface Architecture: Application context, intents, Activity life cycle, multiple screen size s.(2L) User Interface Design: Form widgets, Text Fields, Layouts, Button control, toggle buttons, Spinners(Combo boxes), Images, Menu, Dialog.(2L)
Unit-V
Database: Understanding of SQL database, connecting with the database. (2L)

Suggested readings

Websites
**STRUTS FRAMEWORK**

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

**Scope**

This course motivates the students to design and create applications using Struts framework

**Objectives**

To help students to

- Know the components of Struts Application
- Implement JSP functions using Struts
- Develop web applications using Struts
- understand the Model, View, Controller (MVC) design pattern and how it is applied by Struts Framework
- how to perform client & server side validation using Struts Validator Framework

**Unit-I**


**Unit-II**


**Unit-III**

Unit-IV

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

Unit-V

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

Suggested readings

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

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.

2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.

3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).

4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.

5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.

6. Perform Stack operations using Linked List implementation.

7. Perform Stack operations using Array implementation. Use Templates.

8. Perform Queues operations using Circular Array implementation. Use Templates.

9. Create and perform different operations on Double-ended Queues using Linked List implementation.

10. Write a program to scan a polynomial using linked list and add two polynomial.

11. Write a program to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration

12. Write a program to display Fibonacci series (i)using recursion, (ii) using iteration

13. Write a program to calculate GCD of 2 number (i) with recursion (ii) without recursion

14. Write a program to create a Binary Search Tree and include following operations in tree:
   (a) Insertion (Recursive and Iterative Implementation)
   (b) Deletion by copying
   (c) Deletion by Merging
   (d) Search a no. in BST
   (e) Display its preorder, postorder and inorder traversals Recursively
   (f) Display its preorder, postorder and inorder traversals Iteratively
(g) Display its level-by-level traversals
(h) Count the non-leaf nodes and leaf nodes
(i) Display height of tree
(j) Create a mirror image of tree
(k) Check whether two BSTs are equal or not

15. Write a program to convert the Sparse Matrix into non-zero form and vice-versa.
16. Write a program to reverse the order of the elements in the stack using additional stack.
17. Write a program to reverse the order of the elements in the stack using additional Queue.
18. Write a program to implement Diagonal Matrix using one-dimensional array.
19. Write a program to implement Lower Triangular Matrix using one-dimensional array.
20. Write a program to implement Upper Triangular Matrix using one-dimensional array.
21. Write a program to implement Symmetric Matrix using one-dimensional array.
22. Write a program to create a Threaded Binary Tree as per inorder traversal, and implement operations like finding the successor / predecessor of an element, insert an element, inorder traversal.
23. Write a program to implement various operations on AVL Tree.
1. Write a program (using fork() and/or exec() commands) where parent and child execute:
   a) same program, same code.
   b) same program, different code.
   c) before terminating, the parent waits for the child to finish its task.
2. Write a program to report behavior of Linux kernel including CPU information and memory information.
3. Write a program to print file details including owner access permissions, file access time, where file name is given as argument.
4. Write a program to copy files using system calls.
5. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.
6. Write program to implement FCFS scheduling algorithm.
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.
Instruction Hours / week: L: 0 T: 0 P: 4   Marks: Int : 40 Ext : 60   Total: 100

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.
1. Create ―Hello World‖ application. That will display ―Hello World‖ in the middle of the screen in the emulator. Also display ―Hello World‖ in the middle of the screen in the Android Phone.

2. Create an application with login module. (Check username and password).

3. Create spinner with strings taken from resource folder (res >> value folder) and on changing the spinner value, Image will change.

4. Create a menu with 5 options and and selected option should appear in text box.

5. Create a list of all courses in your college and on selecting a particular course teacher-in-charge of that course should appear at the bottom of the screen.

6. Create an application with three option buttons, on selecting a button colour of the screen will change.

7. Create and Login application as above. On successful login, pop up the message.

8. Create an application to Create, Insert, update, Delete and retrieve operation on the database.
1. Create a Simple Struts Human Resource (HR) Application
2. Create Struts Login Application Using Action form
3. Create a Strut Global Action Forwards and Action Mappings
4. Create and implement Multiple Struts Configuration File
5. Create Struts Application to implement struts Lookup Dispatch Action
6. Create Struts Application to mapping the struts Action Servlet
7. Create a Feedback form to implement struts HTML Tag
8. Create Struts Application to use of action forms for validating user input.
9. Create Struts Applications to implement Struts Logic Tag
10. Create a Struts LOGIC tag library provides tags that are useful in managing conditional generation of output text
Scope
This course sharpens the theoretical, technical, and practical knowledge of software requirements, analysis, design, implementation, verification and validation, and documentation skills of the students.

Objectives
- Apply their knowledge of mathematics, sciences, and computer science to the modeling, analysis, and measurement of software artifacts.
- Analyze, specify and document software requirements for a software system.
- Implement a given software design using sound development practices.
- Verify, validate, assess and assure the quality of software artifacts.
- 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.

Unit-I

Unit-II

Unit-III

Unit-IV
Unit-V


Suggested Readings


Websites

Scope
The scope of this course is to teach the fundamentals of the database systems at a graduate student level. This course also emphasize database concepts, developments, use and management.

Objectives
To teach students
- the fundamentals of database management systems, techniques for the design of databases, and principles of database administration.
- the role and nature of relational database management systems in today's IT environment.
- Translate written business requirements into conceptual entity-relationship data models.
- develop and manage efficient and effective database applications

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

Unit-II
Relation data model: Relational model concepts, relational constraints, relational algebra.

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

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

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


Websites

Scope

This course will introduce the fundamental languages of the web today - including HTML, powerful JavaScript libraries and ASP. Also, provides the interdisciplinary skill necessary to build and maintain a website.

Objectives

To help student

• to write, deploy, debug and run Javascript code in the context of client-side Web pages
• to understand the structure and syntax of programming languages, with a focus on Javascript’s unique capabilities and techniques
• to develop dynamic web applications, create and consume web services
• to understand the 3-tier software architecture

Unit-I


Unit-II

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

Unit-III

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

Unit-IV

Unit-V

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

Suggested Readings

Developers, New Delhi , John Wiley & Sons Inc.
Course Technology Publishing. (Unit- I).
company, NewDelhi.(Unit -II)
Delhi. (Unit- IV & Unit -V).

Web Sites:

1. www.w3schools.com/
2. www.javascriptkit.com
3. www.aspfree.com
4. www.aspnettutorials.com
Semester – IV

17CAU404A         R PROGRAMMING            3H – 3C

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

Scope
This course teaches students how to program in R and how to use R for effective data analysis. This course also covers practical issues in statistical computing.

Objectives
Upon successful completion of this course, students will be able to:
- To acquire the computing tasks such as using conditional processing statements, loops, and writing one's own functions.
- Perform basic and advanced graphing of data.
- Use statistical distribution functions in R.
- Perform basic statistical modeling of data.

Unit-I

Unit-II
Getting Data In and Out of R : Reading and Writing Data - Reading Data Files with read.table() - Reading in Larger Datasets with read.table - Calculating Memory Requirements for R Objects - Using the readr Package - Using Textual and Binary Formats for Storing Data - Using dput() and dump() - Binary Formats - Interfaces to the Outside World: File Connections - Reading Lines of a Text File - Reading from a URL Connection - Subsetting R Objects - Subsetting a Vector - Subsetting a Matrix - Subsetting Lists - Subsetting Nested Elements of a List - Extracting Multiple Elements of a List - Partial Matching - Removing NA Values.
Unit-III
Dates and Times: Dates in R - Times in R - Operations on Dates and Times.
Managing Data Frames with the dplyr package: Data Frames - The dplyr Package - dplyr Grammar - Installing the dplyr package
- select() - filter() - arrange() - rename() - mutate() - group_by()%>%.
Control Structures: if-else - for Loops - Nested for loops - while Loops - repeat Loops - next, break.

Unit-IV
Coding Standards for R: Loop Functions: Looping on the Command Line - lapply() - sapply() - split() - Splitting a Data Frame - tapply - apply() - Col/Row Sums and Means - Other Ways to Apply - mapply() - Vectorizing a Function.

Unit-V
Debugging: Something’s Wrong! - Figuring Out What’s Wrong - Debugging Tools in R.
Using traceback() - Using debug() - Using recover().
Simulation: Generating Random Numbers - Setting the random number seed - Simulating a Linear Model - Random Sampling.

Suggested Readings
Scope
Software testing course provides stakeholders with information about the quality test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

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

Unit- I
Assessing Testing Capabilities and Competencies

Unit -II
Software Testing Fundamentals
Examining the specification: Getting started – Performing a high-level review of the specification – Low-level specification test techniques; Testing the software with blinders on: Dynamic Black-Box Testing- Test-to-Pass and Test-to-Fail- Equivalence Partitioning- Data testing – State testing – Other Black-box test techniques; Examining the code: Static White-Box testing- Formal reviews – Coding Standards and Guidelines- Generic Code Review Checklist; Testing the software with X-Ray glasses: Dynamic
Unit- III

Software Testing Techniques

Determining your software testing techniques: Testing Techniques/Tools selection process – Selecting Techniques/tools – Structural system testing techniques - Functional

system testing techniques – Unit testing technique – Functional testing and analysis – Functional testing – Test factor/Test technique matrix

Testing process


Unit - IV

Testing process

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

Unit- V

Testing Specialized Systems and Applications


Suggested Readings


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

**EMPLOYEE Schema**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>NULL KEY</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eno</td>
<td>Char(3)</td>
<td>NO</td>
<td>PRI</td>
</tr>
<tr>
<td>Ename</td>
<td>Varchar(50)</td>
<td>NO</td>
<td>NIL</td>
</tr>
<tr>
<td>Job_type</td>
<td>Varchar(50)</td>
<td>NO</td>
<td>NIL</td>
</tr>
<tr>
<td>Manager</td>
<td>Char(3)</td>
<td>Yes</td>
<td>FK</td>
</tr>
<tr>
<td>Hire_date</td>
<td>Date</td>
<td>NO</td>
<td>NIL</td>
</tr>
<tr>
<td>Dno</td>
<td>Integer</td>
<td>YES</td>
<td>FK</td>
</tr>
<tr>
<td>Commission</td>
<td>Decimal(10,2)</td>
<td>YES</td>
<td>NIL</td>
</tr>
<tr>
<td>Salary</td>
<td>Decimal(7,2)</td>
<td>NO</td>
<td>NIL</td>
</tr>
</tbody>
</table>

**DEPARTMENT Schema**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>NULL KEY</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dno</td>
<td>Integer</td>
<td>No</td>
<td>PRI</td>
</tr>
<tr>
<td>Dname</td>
<td>Varchar(50)</td>
<td>Yes</td>
<td>NULL</td>
</tr>
<tr>
<td>Location</td>
<td>Varchar(50)</td>
<td>Yes</td>
<td>New Delhi</td>
</tr>
</tbody>
</table>

**Query List**

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than $2850.
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in the range of $1500 and $2850.

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

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

10. Query to display Name and Job of all employees who don't have a current Manager.

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

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

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

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

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

16. Query to display the Current Date.

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

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

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

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

21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.

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

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

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

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

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

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

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

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

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

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

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

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

35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a ‘T’.

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

37. Query to display the department no, name and job for all employees in the Sales department.
Instruction Hours / week: L: 0 T: 0 P: 4   Marks: Int : 40 Ext : 60   Total: 100

1. Develop a website for Karpagam University using HTML.
2. Write Online Quiz program (Include Style Sheets)
3. Create a simple animation using DHTML.
4. Write a program to apply Mask into an Image Using Filters in DHTML.
5. Generate web page that represents clock-every 60 see the page updated with server current time Using JavaScript.
6. Design a form and validate it using JavaScript.
7. Show the demo of AD Rotator Component
8. Write Database Access program using ASP.
9. Program to retrieve Cookies information using ASP
10. Program to count web page hits using ASP
11. Program to create Date & Time, String Manipulation using ASP
12. Write a program to find the visitor’s Browser Type, IP Address and More Information
1. Write a program to demonstrate functions and operators
2. **Vectors**: Grouping values into vectors, then doing arithmetic and graphs with them
3. **Matrices**: Creating and graphing two-dimensional data sets
4. **Summary Statistics**: Calculating and plotting some basic statistics: mean, median, and standard deviation
5. **Factors**: Creating and plotting categorized data
6. **Data Frames**: Organizing values into data frames, loading frames from files and merging them
7. Write a program to design R as a calculator
8. Write a program to demonstrate Probability distributions
9. Write a program to demonstrate Importing and exporting data
10. Write a program to Establish a Regression
1. Create a VB form with the following fields and create the database also for them. Insert 3 records. Using Win Runner tool record the above 3 transaction and test them and produce the Report. (Blackbox Testing).

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

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


5. Write a C program for Boundary Testing.

6. Write a C program for Loop Testing.

7. Write a C program for Integration Testing.

8. Write a C program for Interface Testing.

9. Write a C program for Unit testing.
Scope

This course will turn students into a more skilled technologist and acquire new skills that will position them as a specialized expert in the global IT industry.

Objectives

The educate students to

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

Unit-I

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

Unit-II

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

Unit-III

Other Database Objects - View, Synonyms, Index

Unit-IV

Transaction Control Statements - Commit, Rollback, Savepoint

Unit-V

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


Scope
This course is an introduction to the Python programming language. The course covers data types, control flow, object-oriented programming, and graphical user interface-driven applications.

Objectives
The course educates students on how to:
- use Python interactively
- use Python types, expressions, and None
- use string literals and string type
- use Python statements (if...elif..else, for, pass, continue, . . .)
- utilize high-level data types such as lists and dictionaries

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

Unit-II

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

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

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


Websites

Scope
This course provides an overview of Information Security. Students would be exposed to the spectrum of security activities, methods, methodologies, and.

Objectives
To teach students:

- the basic concepts in information security, including security policies, security models, and security mechanisms.
- the concepts related to applied cryptography, including plain-text, cipher-text, the four techniques for crypto-analysis, symmetric cryptography, and asymmetric cryptography.
- the common vulnerabilities in computer programs.
- different threats and protection to be provided

Unit -I
Cryptography: Substitution ciphers, Transpositions Cipher, Confusion, diffusion.

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

Unit – III
Program Security: Secure programs, Non malicious Program errors, Malicious codes virus, Trap doors, Salami attacks, Covert channels, Control against program.

Unit – IV
Security in Networks: Threats in Networks, Security Controls, firewalls, Intrusion detection systems, Secure e-mails
Unit - V


**Suggested Readings**

Scope
This course introduce students to the basic concepts and techniques of Data Mining, and the data mining software used for solving practical problems.

Objectives
The primary objectives of the course is teach students:
- the algorithms and methods of data mining.
- to develop data mining programs and applications.
- to program using available data mining tools and general-purpose languages.
- the analysis, metrics, visualization and navigation of data mining results.
- to use a few commercial data mining tools

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

Unit-II
**Preparing Data for Mining**: Variable Measures.

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

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

Unit-V
**Data Mining Techniques**: Association Rule Mining, classification and regression techniques, clustering, Scalability and data management issues in data mining algorithms, measures of interestingness
Suggested Readings

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

Scope
The Course gives fundamental idea to different architectures and the programming skills for microprocessor. It gives the overview about microprocessor architecture, pin diagram, Instruction set descriptions, Assembler directives, stack structure, procedures and macros

Objectives
- Understand the internal architecture of 8085 and the memory organization aspects
- Understand the register organization
- Able to develop programs using different class of instructions
- Able to develop programs to interface real time application modules
- 8085 Interrupts & interrupt responses which will be helpful to develop the programming skills using different group of instructions.

Unit – I

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

Unit – III

Unit – IV

Unit – V - Applications


Suggested Readings:

Semester – V

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

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

End Semester Exam: 3 Hours

Scope
This is an introduction course on computer organization, processor and memory concept, peripherals and system architecture.

Objectives
The objective of this course to train student to assemble/setup and upgrade personal computer systems, perform installation, configuration, and upgrading of hardware and software, install/connect associated peripherals and diagnose and troubleshoot in hardware and software and other peripheral equipment.

Unit-I

Unit-II

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

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

Unit-V
Suggested Readings
Scope

This course introduces the fundamental theories and techniques of digital image processing.

Objectives

The students get exposed to:

- different transformations and their applications
- different filters
- compression techniques
- methods of image restoration

Unit-I


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

Unit – IV
FAX compression (CSUITT Group-3 and Group-4), Symbol-based coding, JBIG-2, Bit-plane encoding, Bit-allocation, Zonal Coding, Threshold Coding, JPEG, Lossless predictive coding, Lossy predictive coding, Motion Compensation

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
Scope
This course teaches issues associated with Open Source technologies, characteristics of key Open Source, the nature of Open Source communities and development processes, and the implications for businesses interested in making use of Open Source technologies.

Objectives
The students would
- come to know the importance of Open source in the development of software
- come to know the role of open source in development and distribution of higher quality, better reliability, more flexibility, and lower cost software.
- learn the open source ideals in order to apply those principles

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

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

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

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

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

Suggested Readings

4. James Lee, Brent Ware,2002. Open Source Web Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP, , Publisher: Addison Wesley Date [UNIT IV – V]
1) SQL* formatting commands
2) To create a table, alter and drop table.
3) To perform select, update, insert and delete operation in a table.
4) To make use of different clauses viz where, group by, having, order by, union and intersection,
5) To study different constraints.

[SQL FUNCTION]
6) To use oracle function viz aggregate, numeric, conversion, string function.
7) To understand use and working with joins.
8) To make use of transaction control statement viz rollback, commit and save point.
9) To make views of a table.
10) To make indexes of a table.

[PL/SQL]
11) To understand working with PL/SQL
12) To implement Cursor on a table.
13) To implement trigger on a table
1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user’s choice.

2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria:
   Grade A: Percentage >=80
   Grade B: Percentage>=70 and <80
   Grade C: Percentage>=60 and <70
   Grade D: Percentage>=40 and <60
   Grade E: Percentage<40

3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.

4. Write a program in python to display the first n terms of Fibonacci series.

5. Write a program in python to find factorial of the given number.

6. Write a program in python to find sum of the following series for n terms: $1 - \frac{2}{2!} + \frac{3}{3!} - \frac{4}{4!} + \ldots - \frac{n}{n!}$

7. Write a program in python to calculate the sum and product of two compatible matrices.
1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois
2. Use of Password cracking tools: John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Perform encryption and decryption of Caesar cipher. Write a script for performing these operations.
4. Perform encryption and decryption of a Rail fence cipher. Write a script for performing these operations.
5. Use nmap/zenmap to analyse a remote machine.
6. Use Burp proxy to capture and modify the message.
7. Demonstrate sending of a protected word document.
8. Demonstrate sending of a digitally signed document.
10. Demonstrate use of steganography tools.
11. Demonstrate use of gpg utility for signing and encrypting purposes.
1. Use the following learning schemes, with the default settings to analyze the weather data (in weather.arff). For test options, first choose "Use training set", then choose "Percentage Split" using default 66% percentage split. Report model percent error rate.

2. Using iris dataset preprocess and classify it with J4.8 and Naïve Bayes Classifier. Examine the tree in the Classifier output panel.

3. Using the datasets ReutersCorn-Train and ReutersGrain-Train. Classify articles using binary attributes and word count attributes.

4. Apply any two association rule based algorithm for the supermarket analysis.

5. Using weka Experimenter perform comparison analysis of J48, oneR and ID3 for vote dataset.

6. Using Weka Experimenter perform comparison analysis of Naïve Bayes with different datasets.

7. Apply ZeroR, OneR, and J48, to classify the Iris data in an experiment using 10 train and test runs, with 66% of the data used for training and 34% used for testing.

8. Using Weka Knowledge flow Set up a flow to load an ARFF file (batch mode) and perform a cross-validation using J48 (WEKA’s C4.5 implementation).
Assembly Language Programming

1. Addition of 8/16 bit and array of Data
2. Subtraction of 8/16 bit and array of Data
3. Multiplication of 8-bit Number
4. Division of 8-bit Number Write a program for 32-bit binary division and multiplication
5. Fill and Transfer an Array of Data
6. Ascending and Descending of an array
7. Data Transfer using Parallel Ports
8. Stepper Motor Interface
9. Traffic Light Interface
10. A/D and D/A Converter
1. Identifying External Ports and Interfacing
2. Identifying PC cards and Interfacing.
3. Assembling of PC
4. Preventive Maintenance of a PC
5. Trouble Shooting of SMPS
6. Keyboard Servicing
7. Study of CRT
8. Communication and Bus Interfacing
10. Installing System And Application Software
17CAU514A  DIGITAL IMAGE PROCESSING PRACTICAL  Semester – V

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

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
   c. To write and execute program for wavelet transform on given image and perform
      inverse wavelet transform to reconstruct image.
9. Write and execute programs for image frequency domain filtering
   a. Apply FFT on given image
   b. Perform low pass and high pass filtering in frequency domain
   c. Apply IFFT to reconstruct image
10. Write a program in C and MATLAB/SCILAB for edge detection using
    different edge detection mask
11. Write and execute program for image morphological operations erosion and dilation.
1. Create —Hello World application. That will display —Hello World in the middle of the screen in the emulator. Also display —Hello World in the middle of the screen in the Android Phone.

2. Create an application with login module. (Check username and password).

3. Create spinner with strings taken from resource folder and on changing the spinner value, Image will change.

4. Create a menu with 5 options and selected option should appear in text box.

5. Create a list of all courses in your college and on selecting a particular course teacher-in-charge of that course should appear at the bottom of the screen.

6. Create an application with three option buttons, on selecting a button colour of the screen will change.

7. Create and Login application as above. On successful login, pop up the message.

8. Create an application to Create, Insert, update, Delete and retrieve operation on the database.
Scope

This course targets web developers. This course lays down a strong foundation in PHP scripting language and provides the student with the necessary tools to write scripts that are secure, efficient, and reliable. This course also supplements the necessary skill in designing the structure of the website including the information architecture, the layout or the pages and the conceptual design with branding.

Objectives

The students would be able to

- design and implement a database-driven Web application on the LAMP platform
- work with open source applications that deal with database and website development.
- establish a working environment for PHP web page development

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

UNIT-V

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

Suggested Readings

1. David Sklar, Adam Trachtenberg, (2014). PHP Cookbook: Solutions & Examples for PHP.
Scope

This course teaches the student the concepts and principles that underlie modern operating systems, and a practice component to relate theoretical principles with operating system implementation. Learn about processes and processor management, concurrency and synchronization, memory management schemes, file system and secondary storage management, security and protection, etc.

Objectives

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

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

17CAU602A  CLOUD COMPUTING

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

Scope
This course covers the current cloud computing technologies, including technologies for Infrastructure as a Service, Platform as a Service, Software as a Service, Physical Systems as a Service and cloud security.

Objectives
To provide students:

- a good understanding of the concepts, standards and protocols in Cloud computing
- knowledge about cloud computing Models including Infrastructure/ Platform/ Software as a service
- Public cloud, private cloud and hybrid clouds
- Security and Privacy issues in the Cloud

Unit-I

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


Websites

1. en.wikipedia.org/wiki/Cloud_computing
Scope
Database administration course provides students with skills for installation, configuration and tuning a database and utilizing resources optimally.

Objective
The students becomes an entry-level database administration so that he/she

- manage database services and clients
- implement and configure a database environment
- apply database optimization, maintenance and recovery procedures
- be able to move the data from one database to another database.
- plan and implement a comprehensive backup and recovery strategy
- import, export and transform data from various sources

Unit I

Unit II

Unit III
Unit IV


Unit-V

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

Suggested Readings


Websites:

2. www.oracle-base.com/articles/10g/
3. www.adp-gmbh.ch/ora/misc/10g.html
Scope
This scope of this course to explain the students the fundamentals of big data analytics and the methodologies used in storing, manipulating, and analyzing massive dataset.

Objectives
To impart to students the skills required to design scalable systems that can accept, store, and analyze large volumes of unstructured data.
The objective of this course is to ascertain that the students know the fundamental techniques and tools used to design and analyze large volumes of data.

Unit-I

Unit-II

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

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

Unit-V
Integrating Data Sources-Real-Time Data Streams and Complex Event Processing-Operationalizing Big Data.
Suggested readings


Websites

1. www.oracle.com/BigData
4. www.solacesystems.com
5. en.wikipedia.org/wiki/Big_data
**Scope**

This course enables for good understanding of the role of system programming and the scope of duties and tasks of a system programmer. This course enables to learn the concepts and principles of developing system-level software (e.g., compiler, and networking software)

**Objectives**

- To introduce students the concepts and principles of system programming
- To provide students the knowledge about both theoretical and practical aspects of system programming, teaching them the methods and techniques for designing and implementing system-level programs.

**Unit-I**

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

**Unit-II**

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

**Unit-III**

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

**UNIT-IV**

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

**UNIT-V**

**Code Generation:** Object code generation

**Suggested Readings**

1. Create a PHP page using functions for comparing three integers and print the largest number.

2. Write a function to calculate the factorial of a number (non-negative integer). The function accept the number as an argument.

3. Write a simple PHP program to check whether the given number is prime or not.

4. Create a PHP page which accepts string from user. After submission that page displays the reverse of provided string.

5. Write a PHP function that checks if a string is all lower case.

6. Write a PHP script that checks whether a passed string is palindrome or not? (A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run)

7. WAP to sort an array.

8. Write a PHP script that removes the whitespaces from a string.
Sample string: 'The quick " " brown fox' Expected Output: Thequick""brownfox

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

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

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

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

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

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

14. Write a simple PHP program to check that emails are valid.
15. Write a simple PHP program to print first n even numbers.
16. $color = array('white', 'green', 'red')
   
   Write a PHP script which will display the colors in the following way : Output :
   white, green, red,
17. Using switch case and dropdown list display a —Hello‖ message depending on the
   language selected in drop down list.
18. Write a PHP program to print 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.
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 aCSUept 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
1. Create virtual machines that aCSUess different programs on same platform.
2. Create virtual machines that aCSUess 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
1. Demo for Globalization Support
2. Setup Listener Security
3. Configuring Recovery Manager
4. Write a program Using Recovery Manager
5. Write a program for Managing Diagnostic Sources
6. Implement Database Recovery
7. Demo for Flashback Database
8. Implement User Error Recovery
9. Write a program for Dealing with Corruption
10. Show the demo for Automated Management
11. Creating a database and do the manipulation.
12. Managing index tables
1. Implement a quicksort using scala.
2. Implement a auction service using scala.
3. Write a scala function to perform any 10 arithmetic operations.
4. Write a program to find the factorial of a given number using recursion.
5. Write a program for string manipulations.
6. Write a program for alphabetic order arrangement of a set of names.
7. Write a program for student records using scala list.
8. Implement any 5 map methods for maintaining customer details.
9. Implement employee records using Files.
10. Write a program to copy the files using command line arguments.
1. To implement an assembler for a hypothetical language.

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

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