B.Sc. COMPUTER SCIENCE
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)
Pollachi Main Road, Eachanari (Post), Coimbatore – 641 021, Tamil Nadu, India
Phone: 0422 – 2980011 –15 Fax No: 0422 – 2980022-23
Email: info@karpagam.com Web: www.kahedu.edu.in
17LSU101

(10)

(10)

(10)

(10)

1.  
2.  
3.  
4.  
5.  

1.  
2.  
3.  

4. 17CSU101 PROGRAMMING FUNDAMENTALS USING C / C++  4H – 4C

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

SCOPE
This course provides student with a comprehensive study of the fundamentals of C and C++ programming language. Classroom lectures stress the strength of C, which provide programmers with the means of writing efficient, maintainable and portable code.

OBJECTIVES
• Know the basic concept of computers
• Understand the concept of a program (i.e., a computer following a series of instructions)
• Understand the concept of a loop – that is, a series of statements which is written once but executed repeatedly- and how to use it in a programming language
• Be able to break a large problem into smaller parts, writing each part as a module or function
• Understand the concept of a program in a high-level language being translated by a compiler into machine language program and then executed.

UNIT-I
Introduction to C and C++:
History of C and C++, Overview of Procedural Programming and Object-Orientation Programming, Using main() function, Compiling and Executing Simple Programs in C++. Data Types, Variables, Constants, Operators and Basic I/O: Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putcharetc), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.hetc). 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, ACSUessing 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 ifstream header file, ifstream, ofstream and fstream classes), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random aCSUess 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 ACSUess, 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


WEB SITES
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.

OBJECTIVE
- 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 Memory and Input-Output Organization


Suggested Readings:

SCOPE
This course provides student with a comprehensive study of the fundamentals of computers.

OBJECTIVES
- Know the basic concept of computers

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

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

UNIT-III
Memory: Primary, secondary, auxiliary memory, RAM, ROM, cache memory, hard disks, optical disks.

UNIT-IV
Computer Organisation and Architecture: C.P.U., registers, system bus, main memory unit, cache memory, Inside a computer, SMPS, Motherboard, Ports and Interfaces, expansion cards, ribbon cables, memory chips, processors.

UNIT-V
Overview of Emerging Technologies: Bluetooth, cloud computing, big data, data mining, mobile computing and embedded systems.

Suggested Readings:

17CSU111 PROGRAMMING FUNDAMENTALS USING C / C++ - PRACTICAL

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

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first n terms of the following series S = 1+1/2+1/3+1/4+……
4. WAP 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. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. WAP to print a triangle of stars as follows (take number of lines from user):

   *
   ***
   *****
   ********
   *********
10. WAP to perform following actions on an array entered by the user:
    i) Print the even-valued elements
ii) Print the odd-valued elements

iii) Calculate and print the sum and average of the elements of array

iv) Print the maximum and minimum element of array

v) Remove the duplicates from the array

vi) Print the array in reverse order

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

11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.

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

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

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

16. Write a menu driven program to perform following operations on strings:
   a) Show address of each character in string

   b) Concatenate two strings without using strcat function.

   c) Concatenate two strings using strcat function.

   d) Compare two strings

   e) Calculate length of the string (use pointers)

   f) Convert all lowercase characters to uppercase

   g) Convert all uppercase characters to lowercase

   h) Calculate number of vowels

   i) Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.

18. WAP to display Fibonacci series (i) using recursion, (ii) using iteration

19. WAP to calculate Factorial of a number (i) using recursion, (ii) using iteration

20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.

21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation):
   a) Sum b) Difference c) Product d) Transpose

22. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).

23. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.

24. Create a class Box containing length, breadth and height. Include following methods in it:
   a) Calculate surface Area
   b) Calculate Volume
   c) Increment, Overload ++ operator (both prefix & postfix)
   d) Decrement, Overload -- operator (both prefix & postfix)
   e) Overload operator == (to check equality of two boxes), as a friend function
   f) Overload Assignment operator
   g) Check if it is a Cube or cuboid
   Write a program which takes input from the user for length, breadth and height to test the above class.

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

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

   Roll No. Name Marks

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

28. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
29. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

17CSU112 COMPUTER SYSTEM ARCHITECTURE - PRACTICAL  
Semester – I  
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
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.

<table>
<thead>
<tr>
<th>Salesperson</th>
<th>Dolls</th>
<th>Trucks</th>
<th>Puzzles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kennedy, Sally</td>
<td>1327</td>
<td>1423</td>
<td>1193</td>
</tr>
<tr>
<td>White, Pete</td>
<td>1421</td>
<td>3863</td>
<td>2934</td>
</tr>
<tr>
<td>Pillar, James</td>
<td>5214</td>
<td>3247</td>
<td>5467</td>
</tr>
<tr>
<td>York, George</td>
<td>2190</td>
<td>1278</td>
<td>1928</td>
</tr>
<tr>
<td>Banks, Jennifer</td>
<td>1201</td>
<td>2528</td>
<td>1203</td>
</tr>
<tr>
<td>Atwater, Kelly</td>
<td>4098</td>
<td>3079</td>
<td>2067</td>
</tr>
<tr>
<td>Pillar, James</td>
<td>5214</td>
<td>3247</td>
<td>5467</td>
</tr>
<tr>
<td>York, George</td>
<td>2190</td>
<td>1278</td>
<td>1928</td>
</tr>
<tr>
<td>Banks, Jennifer</td>
<td>1201</td>
<td>2528</td>
<td>1203</td>
</tr>
<tr>
<td>Atwater, Kelly</td>
<td>4098</td>
<td>3079</td>
<td>2067</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>State</th>
<th>Qtr1</th>
<th>Qtr2</th>
<th>Qtr3</th>
<th>QTR4</th>
<th>Qtr Total</th>
<th>Rate Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delhi</td>
<td>2020</td>
<td>2400</td>
<td>2100</td>
<td>3000</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Punjab</td>
<td>1100</td>
<td>1300</td>
<td>1500</td>
<td>1400</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>U.P.</td>
<td>3000</td>
<td>3200</td>
<td>2600</td>
<td>2800</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Haryana</td>
<td>1800</td>
<td>2000</td>
<td>2200</td>
<td>2700</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Rajasthan</td>
<td>2100</td>
<td>2000</td>
<td>1800</td>
<td>2200</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL**

**AVERAGE**

(a) Apply Formatting as follow:

i. Title in TIMES NEW ROMAN

ii. Font Size - 14

iii. Remaining text - ARIAL, Font Size -10

iv. State names and Qtr. Heading Bold, Italic with Gray Fill Color.

v. Numbers in two decimal places.

vi. Qtr. Heading in center Alignment.
vii. Apply Border to whole data.

(b) Calculate State and Qtr. Total

(c) Calculate Average for each quarter

(d) Calculate Amount = Rate * Total.

2. Given the following worksheet

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll No.</td>
<td>Name</td>
<td>Marks</td>
<td>Grade</td>
</tr>
<tr>
<td>1</td>
<td>1001</td>
<td>Sachin</td>
<td>99</td>
</tr>
<tr>
<td>2</td>
<td>1002</td>
<td>Sehwag</td>
<td>65</td>
</tr>
<tr>
<td>3</td>
<td>1003</td>
<td>Rahul</td>
<td>41</td>
</tr>
<tr>
<td>4</td>
<td>1004</td>
<td>Sourav</td>
<td>89</td>
</tr>
<tr>
<td>5</td>
<td>1005</td>
<td>Har Bhajan</td>
<td>56</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>If Marks</th>
<th>Then Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;= 80</td>
<td>A+</td>
</tr>
<tr>
<td>&gt;= 60 &lt; 80</td>
<td>A</td>
</tr>
<tr>
<td>&gt;= 50 &lt; 60</td>
<td>B</td>
</tr>
<tr>
<td>&lt; 50</td>
<td>F</td>
</tr>
</tbody>
</table>

3. Given the following worksheet

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

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

<table>
<thead>
<tr>
<th>If Total Sales</th>
<th>Commission</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The total sales is sum of sales of all the four quarters.

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

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

**Deductions**
- Provident Fund 6% of Basic
- Group Insurance Premium Rs. 40/- if Basic is \( \leq 1500 \)
  - Rs. 60/- if Basic \( > 1500 \) & Basic \( \leq 3000 \)
  - Rs. 80/- if Basic \( > 3000 \)

Calculate the following:
- Gross Salary = Basic + HRA + DA + Conveyance + Entertainment
- Total deduction = Provident Fund + Group Insurance Premium

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

<table>
<thead>
<tr>
<th>No. of Installments</th>
<th>5%</th>
<th>6%</th>
<th>7%</th>
<th>8%</th>
<th>9%</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
</tr>
<tr>
<td>4</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
</tr>
<tr>
<td>5</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
</tr>
<tr>
<td>6</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
</tr>
</tbody>
</table>

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

\[
\text{Interest} = \text{Principal} \times \text{Rate} \times \text{Time}
\]
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.

<table>
<thead>
<tr>
<th>Salesman</th>
<th>2000</th>
<th>2001</th>
<th>2002</th>
<th>2003</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>10000</td>
<td>12000</td>
<td>20000</td>
<td>50000</td>
</tr>
<tr>
<td>S2</td>
<td>15000</td>
<td>18000</td>
<td>50000</td>
<td>60000</td>
</tr>
<tr>
<td>S3</td>
<td>20000</td>
<td>22000</td>
<td>70000</td>
<td>70000</td>
</tr>
<tr>
<td>S4</td>
<td>30000</td>
<td>30000</td>
<td>10000</td>
<td>80000</td>
</tr>
<tr>
<td>S5</td>
<td>40000</td>
<td>45000</td>
<td>12500</td>
<td>90000</td>
</tr>
</tbody>
</table>

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

8. Enter the following data in Excel Sheet

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

<table>
<thead>
<tr>
<th>EXPENSES</th>
<th>JAN</th>
<th>FEB</th>
<th>MARCH</th>
<th>QUARTER TOTAL</th>
<th>QUARTER AVERAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rent</td>
<td>600.00</td>
<td>600.00</td>
<td>600.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Telephone</td>
<td>48.25</td>
<td>43.50</td>
<td>60.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Utilities</td>
<td>67.27</td>
<td>110.00</td>
<td>70.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Credit Card</td>
<td>200.00</td>
<td>110.00</td>
<td>70.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oil</td>
<td>100.00</td>
<td>150.00</td>
<td>90.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AV to Insurance</td>
<td>150.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cable TV</td>
<td>40.75</td>
<td>40.75</td>
<td>40.75</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

9. Enter the following data in Excel Sheet

**TOTAL REVENUE EARNED FOR SAM’S BOOKSTALL**

<table>
<thead>
<tr>
<th>Publisher name</th>
<th>1997</th>
<th>1998</th>
<th>1999</th>
<th>2000</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Rs.1000.00</td>
<td>Rs.1100.00</td>
<td>Rs.1300.00</td>
<td>Rs.800.00</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Rs.1500.00</td>
<td>Rs.700.00</td>
<td>Rs.1000.00</td>
<td>Rs.2000.00</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Rs.700.00</td>
<td>Rs.900.00</td>
<td>Rs.1500.00</td>
<td>Rs.600.00</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Rs.1200.00</td>
<td>Rs.500.00</td>
<td>Rs.200.00</td>
<td>Rs.1100.00</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Rs.800.00</td>
<td>Rs.1000.00</td>
<td>Rs.3000.00</td>
<td>Rs.560.00</td>
<td></td>
</tr>
</tbody>
</table>

(a) Compute the total revenue earned.
(b) Plot the line chart to compare the revenue of all publisher for 4 years.
(c) Chart Title should be “Total Revenue of Sam’s Bookstall (1997-2000)‘
(d) Give appropriate categories and value axis title.

10. Generate 25 random numbers between 0 & 100 and find their sum, average and count. How many no. are in range 50-60.
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 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 IV: Environmental Pollution - Definition, Causes, effects and control measures of Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, Nuclear hazards, Solid waste management: Causes, effects and control measures of urban and industrial wastes. Role of an individual in prevention of pollution. Pollution case studies. Disaster management: Foods, earthquake, cyclone and landslides.


Suggested Readings


17LSU201:  

1.  

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.
II).  ஊலுக்கைக்கு நூற்றாண்டு முதல் முதல் வரை – 'பாணியாளர் சமையல்'  

'சின்னானாய்க்கு' சான்றல் நூற்றாண்டு முதல் வரை: 218-249. 

'சின்னானாய்க்கு' சான்றல் நூற்றாண்டு முதல் வரை: 286-295.

III : (6)  

II).  ஊலுக்கைக்கு நூற்றாண்டு முதல் முதல் வரை: (21-29) – 'பாணியாளர் சமையல்': 

'சின்னானாய்க்கு' சான்றல் நூற்றாண்டு முதல் வரை: 

'சின்னானாய்க்கு' சான்றல் நூற்றாண்டு முதல் வரை: 207-234. 

'சின்னானாய்க்கு' சான்றல் நூற்றாண்டு முதல் வரை: 482-485. 

IV : (10)  

II).  ஊலுக்கைக்கு நூற்றாண்டு முதல் முதல் வரை: 

'சின்னானாய்க்கு' சான்றல் நூற்றாண்டு முதல் வரை: 

'சின்னானாய்க்கு' சான்றல் நூற்றாண்டு முதல் வரை: 

'சின்னானாய்க்கு' சான்றல் நூற்றாண்டு முதல் வரை: 

'சின்னானாய்க்கு' சான்றல் நூற்றாண்டு முதல் வரை:
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

17ENU201 ENGLISH

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

(For all undergraduate students admitted from 2016 onwards)
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 - 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 – II

17CSU201 PROGRAMMING IN JAVA 4H – 4C

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

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
- know the java path setting and programming techniques
- basic java programming and Applet programming
- understand the fundamental of Packages and aCSUess 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 ACSUess 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. ACSUessing 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:

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

<table>
<thead>
<tr>
<th>17CSU202</th>
<th>DISCRETE STRUCTURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester – II</td>
<td></td>
</tr>
<tr>
<td>4H – 4C</td>
<td></td>
</tr>
</tbody>
</table>

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.

Objective: 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

TEXT BOOK

REFERENCES


17CSU203      MULTIMEDIA AND APPLICATIONS      Semester – II

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

SCOPE
Understand basic multimedia concepts, devices and the current trends in multimedia. Has the ability to build a multimedia project.

OBJECTIVES
• Understand basic multimedia concepts.
• Acquire basic knowledge on Multimedia devices.
• Understand current trends in multimedia by experiencing a variety of applications and development packages.
• Be able to design different application in M.M and use different tools like adobe Photoshop and macromedia flash.

UNIT-I
**Multimedia:** Introduction to multimedia, components, uses of multimedia, 6L multimedia applications, virtual reality. **Text:** Fonts & Faces, Using Text in Multimedia, Font Editing & Design Tools, Hypermedia & Hypertext.

**UNIT-II**

**Images:** Still Images – bitmaps, vector drawing, 3D drawing & rendering, natural light & colors, computerized colors, color palettes, image file formats. **Sound:** Digital Audio, MIDI Audio, MIDI vs Digital Audio, Audio File Formats.

**UNIT-III**

**Video:** How video works, analog video, digital video, video file formats, video shooting and editing. **Animation:** Principle of animations, animation techniques,

**UNIT-IV**

**Animation:** animation file formats. **Internet and Multimedia:** www and HTML, multimedia on the web – web servers, 6L web browsers, web page makers and site builders.

**UNIT-V**

**Making Multimedia:** Stages of a multimedia project, Requirements to make good multimedia, Multimedia Hardware - Macintosh and Windows production Platforms, Hardware peripherals - Connections, Memory and storage devices, Multimedia software and Authoring tools.

**Suggested Readings:**

**WEB SITES**
1. en.wikipedia.org/wiki/Multimedia
2. www.arena-multimedia.com/
3. www.nextwavemultimedia.com/
1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of .length in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from
   the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and
   StringBuffer class like setCharAt(), setLength(), append(), insert(), concatenation 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 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
   \[ \frac{1}{1\times2} + \frac{1}{2\times3} + \ldots + \frac{1}{n(n+1)} = \frac{n}{n+1} \]
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
9. Write a C Program to construct the truth table for the following formula.
   (i) \( P \land Q \land R \)   (ii) \( P \lor Q \land R \)   (iii) \( P \land Q \land R \)

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

17CSU213  MULTIMEDIA AND APPLICATIONS - PRACTICAL  4H – 2C

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

Practical exercises based on concepts listed in theory using Flash/ GIMP/ PhotoShop/ Animation Tools/ Image Editors/ Video Editors.

Implement the followings using Flash-

1. Create an animation using the tools panel and the properties panel to draw the following – Line, pe, oval, circle, rectangle, square, pencil, brush, lasso tool.
2. Create an animation using text tool to set the font, size, color etc.
3. Create an animation using Free transform tool that should use followings- Move Objects, Skew Objects, Stretch Objects, Rotate Objects, Stretch Objects while maintaining proportion, Rotate Objects after relocating the center dot.
4. Create an animation using layers having following features- Insert layer, Delete layer, guide layer, Mask layer.
5. Modify the document (changing background color etc.) using the following tools Eraser tool, Hand tool, Ink bottle tool, Zoom tool, Paint Bucket tool, Eyedropper tool.
6. Create an animation for bus car race in which both starts from the same point and car wins the race.
7. Create an animation in which text Hello gets converted into GoodBye (using motion/shape tweening).
8. Create an animation having five images having fade-in fade-out effect.
9. Create an scene to show the sunrise (using multiple layers and motion tweening).
10. Create an animation to show the ripple effect.
11. Create an animation (using Shape tweening and shape hints) for transforming one shape into another.
12. Create an animation for bouncing ball (you may use motion guide layer).

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
- possess intermediate level problem solving and algorithm development skills on the computer
- be able to analyze algorithms using big-Oh notation
- understand the fundamental data structures such as lists, trees, and graphs
- understand the fundamental algorithms such as searching, and sorting

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


WEB SITES
http://www.cs.sunysb.edu/~skiena/214/lectures/
www.amazon.com/Teach-Yourself-Structures-Algorithms

17CSU302 OPERATING SYSTEMS Semester – III 4H – 4C

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

SCOPE
This course recognize the concepts and principles of operating systems, provide students with the basic knowledge and skills of memory, device and Process management and techniques and provide experience on MS Windows and LINUX environment.

OBJECTIVES

- Explain basic Idea about the operating system.
- Concept and techniques involved in memory, device and Process management.
- Work in MS Windows and LINUX environment.

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

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/
17CSU303  COMPUTER NETWORKS  4H – 4C

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

SCOPE
This course is to master the fundamentals of data communications networks by gaining a working knowledge of data transmission concepts, understanding the operation of all seven layers of OSI Model and the protocols used in each layer.

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 ACSUess 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 ACSUess 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**

**WEB SITES**
1. en.wikipedia.org/wiki/Internet_protocol_suite  
3. www.yale.edu/pclt/COMM/TCPIP.HTM  
4. www.w3schools.com/tcpip/default.asp
SCOPE
This course motivates the students to design, create, deploy, and test applications for the Android mobile phone platform.

OBJECTIVES
- Student can build own Android apps.
- Explain the differences between Android and other mobile development environments.
- Understand how Android applications work, their life cycle, manifest, Intents, and using external resources.
- Design and develop useful Android applications with compelling user interfaces by using, extending, and creating your own layouts and Views and using Menus.
- Take advantage of Android's APIs for data storage, retrieval, user preferences, files, databases, and content providers.

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 SQLite database, connecting with the database. (2L)

Suggested Readings

REFERENCES
SCOPE
A student who successfully completes this course should be able to learn how to use MATLAB, learn how to program in MATLAB, ability to create a computer program to solve problems in science and engineering.

OBJECTIVE

- To learn fundamental programming concepts using a block-structured language (MATLAB).
- To learn General problem-solving techniques, including the concept of step-wise refinement applied to the development of algorithms.

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

UNIT-II
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**


---

**17CSU311 DATA STRUCTURES - PRACTICAL**

**Semester – II**

**4H – 2C**

**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. 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 fibonaCSUi series (i) using recursion, (ii) using iteration
13. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion
14. WAP to create a Binary Search Tree and include following operations in tree: (a) Insertion (Recursive and Iterative Implementation)
   (b) Deletion by copying
   (c) Deletion by Merging
   (d) Search a no. in BST
   (e) Display its preorder, postorder and inorder traversals Recursively
   (f) Display its preorder, postorder and inorder traversals Iteratively
   (g) Display its level-by-level traversals
   (h) Count the non-leaf nodes and leaf nodes
   (i) Display height of tree
   (j) Create a mirror image of tree
   (k) Check whether two BSTs are equal or not
15. WAP to convert the Sparse Matrix into non-zero form and vice-versa.
16. WAP to reverse the order of the elements in the stack using additional stack.
17. WAP to reverse the order of the elements in the stack using additional Queue.
18. WAP to implement Diagonal Matrix using one-dimensional array.
19. WAP to implement Lower Triangular Matrix using one-dimensional array.
20. WAP to implement Upper Triangular Matrix using one-dimensional array.
21. WAP to implement Symmetric Matrix using one-dimensional array.
22. WAP to create a Threaded Binary Tree as per inorder traversal, and implement operations like finding the successor / predecessor of an element, insert an element, inorder traversal.
23. WAP to implement various operations on AVL Tree.
17CSU312 OPERATING SYSTEMS - PRACTICAL 4H – 2C

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

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.

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. 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 ―identity‖ 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 \(A=[1 2; -1 0]\) the identity matrix \(I=[1 0; 0 1]\) 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 \(1, 4, 9, 16, \ldots, \sqrt{N}\)th 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:

\[
\begin{align*}
\rho_n &= \sqrt{n} \\
\theta_n &= \frac{137.51}{180} \pi n
\end{align*}
\]

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: \(1f, 3f, 5f, \ldots\). The amplitude of the \(n\)th 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

17CSU401 DESIGN AND ANALYSIS OF ALGORITHMS 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
• possess intermediate level problem solving and algorithm development skills on the computer
• be able to analyze algorithms using big-Oh notation
• understand the fundamental data structures such as lists, trees, and graphs
• understand the fundamental algorithms such as searching, and sorting

UNIT-I
UNIT-II
**Sorting and Searching Techniques:** Elementary sorting techniques–Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques - Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Searching Techniques, Medians & Order Statistics, complexity analysis;

UNIT-III
**Lower Bounding Techniques:** Decision Trees **Balanced Trees:** Red-Black Trees

UNIT-IV
**Advanced Analysis Technique:** Amortized analysis **Graphs:** Graph Algorithms–Breadth First Search, Depth First Search and its Applications, Minimum Spanning Trees.

UNIT-V
**String Processing:** String Matching, KMP Technique.

**Suggested Readings**

SCOPE
The graduates of the software engineering program shall be able to apply proper theoretical, technical, and practical knowledge of software requirements, analysis, design, implementation, verification and validation, and documentation. This course enables the students to resolve conflicting project objectives considering viable tradeoffs within limitations of cost, time, knowledge, existing systems, and organizations.

OBJECTIVES
- Apply their knowledge of mathematics, sciences, and computer science to the modeling, analysis, and measurement of software artifacts.
- Work effectively as leader/member of a development team to deliver quality 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.
• 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

UNIT-II:

UNIT-III:

UNIT-IV:

UNIT-V

Suggested Readings

SCOPE
The Objective of Database Management System includes learning of relational data model, database design and Transaction model.

OBJECTIVES
- Understand the role and nature of relational database management systems in today's IT environment.
- Translate written business requirements into conceptual entity-relationship data models.

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: SQLqueries 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.

References:

WEB SITES
SCOPE
This course design focuses on the structure of the website including the information architecture, the layout or the pages and the conceptual design with branding. PHP helps the students for developing dynamic web pages.

OBJECTIVES
- understand the fundamentals of HTML and use different formatting options
- creation of tables and frames
- relate with DHTML and CSS
- To work with open source applications that deal with database and website development.

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


WEB SITES

1. www.w3schools.com/
2. alexle.net/archives/category/web-technolgy
3. jmarshall.com/easy/
4. www.php.net/
5. en.wikipedia.org/wiki/PHP
6. www.w3schools.com/PHP/DEfaULT.asP
SCOPE
This course relates to the interface between web servers and their clients. The course provides information includes markup languages, programming interfaces and languages, and standards for document identification and display. The use of Web technology makes to enhance active student learning and improves their creativity in making web pages.

OBJECTIVES

- To Create a new webpage
- To understand the fundamental features of web applications.
- To understand the objects and components needed for a web designing.

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

UNIT-II

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

2. Michael, J. Young. Step by Step XML.

17CSU411       DESIGN AND ANALYSIS OF ALGORITHMS - PRACTICAL

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

1. Implement Insertion Sort (The program should report the number of comparisons) ii.
   Implement Merge Sort(The program should report the number of comparisons)

2. Implement Heap Sort(The program should report the number of comparisons)

3. Implement Randomized Quick sort (The program should report the number of comparisons)

4. Implement Radix Sort

5. Create a Red-Black Tree and perform following operations on it: i. Insert a node
   ii. Delete a node
   iii. Search for a number & also report the color of the node containing this number.

6. Write a program to determine the LCS of two given sequences

7. Implement Breadth-First Search in a graph

8. Implement Depth-First Search in a graph
9. Write a program to determine the minimum spanning tree of a graph

For the algorithms at S.No 1 to 3 test run the algorithm on 100 different inputs of sizes varying from 30 to 1000. Count the number of comparisons and draw the graph. Compare it with a graph of nlogn.
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

### 17CSU413  DATA BASE MANAGEMENT SYSTEMS - PRACTICAL

**Semester – IV**

**4H – 2C**

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

**Marks:** Int : 40 Ext : 60  **Total:** 100

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>
</table>

---

*Bachelor of Science Computer Science, 2017. Karpagam Academy of Higher Education, Coimbatore India – 641 021.  # 62*
<table>
<thead>
<tr>
<th>Dno</th>
<th>Integer</th>
<th>No PRI</th>
<th>NULL</th>
</tr>
</thead>
<tbody>
<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.

17CSU414A                      HTML PROGRAMMING - PRACTICAL          Semester – IV
                                   3H – 1C

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

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 linking as well as External linking.

4. Create a table using HTML which consists of columns for Roll No., Student's name and grade.
5 Create a Table with the following view:

<table>
<thead>
<tr>
<th>Roll No.</th>
<th>Name</th>
<th>Grade</th>
<th></th>
<th>Place an image here</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6 Create a form using HTML which has the following types of controls:
   I. Text Box
   II. Option/radio buttons
   III. Check boxes
   IV. Reset and Submit buttons

Q.7 Create HTML documents (having multiple frames) in the following three formats:

```
Frame1
Frame2
Frame1
  Frame2
  Frame3
```
Exercise #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.

Exercise 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:

```xml
<book>
  <coverInfo>
    <title>The XML Handbook</title>
    <author>Charles F. Goldfarb</author>
  </coverInfo>
</book>
```
Exercise #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.

Exercise #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.

Exercise #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.
SCOPE
This course provides an overview of Information Security and Assurance. Students will be exposed to the spectrum of security activities, methods, methodologies, and procedures with emphasis on practical aspects of Information Security.

OBJECTIVES
- State the basic concepts in information security, including security policies, security models, and security mechanisms.
- Explain concepts related to applied cryptography, including plain-text, cipher-text, the four techniques for crypto-analysis, symmetric cryptography, asymmetric cryptography, digital signature, message authentication code, hash functions, and modes of encryption operations.
- Explain common vulnerabilities in computer programs, including buffer overflow vulnerabilities, time-of-check to time-of-use flaws, incomplete mediation.

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. Threats: Protection in OS: Memory and Address Protection, ACSUess control, File Protection, User Authentication.

Unit – IV


Unit V

Administrating Security


Suggested Readings

SCOPE
This course is to master the fundamentals of communications networks by gaining a working knowledge of data transmission concepts, understanding the operation of all seven layers of OSI Model and the protocols used in each layer.

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.
- TCP, UDP, SCTP protocols Ethernet and LAN administration.
- Details of IP operations in the INTERNET and associated routing principles

UNIT-I

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


Semester – V

17CSU502A MICROPROCESSOR 4H – 4C

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

Unit – I- Introduction to Microprocessor

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

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

Unit – IV - Peripheral Devices

Unit – V - Applications

Suggested Readings:

SCOPE
The objectives of this course are to make the students learn the fundamental theories and techniques of digital image processing, cover the fundamental concepts of visual perception and image acquisition, basic techniques of image manipulation, segmentation and coding, and a preliminary understanding of Computer Vision.

OBJECTIVES
- To perform image manipulations and analysis in many different fields.
- To provide students with the ability to apply knowledge of computing, mathematics, science and engineering to solve problems in multidisciplinary research.

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:**

**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
The course aims to provide an introduction to the basic principles, techniques, and applications of Machine Learning.

OBJECTIVES
- Students will have a broad understanding of machine learning algorithms and their use in data-driven knowledge discovery and program synthesis.
- Students will have designed and implemented several machine learning algorithms in Java.
- Students will also be able to identify, formulate and solve machine learning problems that arise in practical applications.
- Students will have a knowledge of the strengths and weaknesses of different machine learning algorithms (relative to the characteristics of the application domain) and be able to adapt or combine some of the key elements of existing machine learning algorithms to design new algorithms as needed.
UNIT-I
Introduction: Concept of Machine Learning, Applications of Machine Learning, Key elements of Machine Learning, Supervised vs. Unsupervised Learning, Statistical Learning: Bayesian Method, The Naive Bayes Classifier

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

UNIT-III
Linear Regression: Prediction using Linear Regression, Gradient Descent, Linear Regression with one variable, Linear Regression with multiple variables, Polynomial Regression, Feature Scaling/Selection.
Logistic Regression: Classification using Logistic Regression, Logistic Regression vs. Linear Regression, Logistic Regression with one variable and with multiple variables.

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

UNIT-V

Suggested Readings
17CSU503B  INTRODUCTION TO DATA SCIENCES  4H – 4C

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

SCOPE

This course gives an introduction to the basics of data sciences and leave armed with practical experience extracting value from big data.

OBJECTIVES

- Building a comprehensive working knowledge and expertise around various analytical and database tools which is a key step to excel in Big Data and Data Science fields.
- The Data Science course covers topics in a comprehensive manner with applications of R programming.

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 RStudio.

UNIT-II

R Programming Basics: Overview of R, R data types 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 APIs, from databases 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 modeling 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 analyses 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


2. Foster Provost, & Tom Fawcett. (2013). Data Science for Business What You Need to Know About Data Mining and Data-Analytic Thinking. O'Reilly.


SCOPE
The Objective of Relational Database Management System including relational, object-relational, and object-oriented systems, SQL standards, algebraic query languages, integrity constraints, triggers, functional dependencies, and normal forms. Other topics include tuning database transactions, security from the application perspective, and data warehousing.

OBJECTIVES

- Understand the role and nature of relational database management systems (RDBMS) in today's IT environment.
- Translate written business requirements into conceptual entity-relationship data models.
- Convert conceptual data models into relational database schemas using the SQL Data Definition Language (DDL).
- Query and manipulate databases using the SQL Data Manipulation Language (DML).

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 programming language is versatile, robust and comprehensive programming language. It has true portability features and can be used across a multitude of platforms.

OBJECTIVES

- Master the principles of object-oriented programming and the interplay of algorithms and data structures in well-written modular code;
- Solve problems requiring the writing of well-documented programs in the Python language, including use of the logical constructs of that language;
- Demonstrate significant experience with the Python program development environment.

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-Identifier and keywords-Literals-Strings-Operators(Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bitwise 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

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. 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.
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. 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
   c. 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.
1. Perform elementary mathematical operations in Octave/MATLAB like addition, multiplication, division and exponentiation.

2. Perform elementary logical operations in Octave/MATLAB (like OR, AND, Checking for Equality, NOT, XOR).

3. Create, initialize and display simple variables and simple strings and use simple formatting for variable.

4. Create/Define single dimension / multi-dimension arrays, and arrays with specific values like array of all ones, all zeros, array with random values within a range, or a diagonal matrix.

5. Use command to compute the size of a matrix, size/length of a particular row/column, load data from a text file, store matrix data to a text file, finding out variables and their features in the current scope.

6. Perform basic operations on matrices (like addition, subtraction, multiplication) and display specific rows or columns of the matrix.

7. Perform other matrix operations like converting matrix data to absolute values, taking the negative of matrix values, adding/removing rows/columns from a matrix, finding the maximum or minimum values in a matrix or in a row/column, and finding the sum of some/all elements in a matrix.

8. Create various type of plots/charts like histograms, plot based on sine/cosine function based on data from a matrix. Further label different axes in a plot and data in a plot.

9. Generate different subplots from a given plot and color plot data.

10. Use conditional statements and different type of loops based on simple example/s.

11. Perform vectorized implementation of simple matrix operation like finding the transpose of a matrix, adding, subtracting or multiplying two matrices.

12. Implement Linear Regression problem. For example, based on a dataset comprising of existing set of prices and area/size of the houses, predict the estimated price of a given house.

13. Based on multiple features/variables perform Linear Regression. For example, based on a number of additional features like number of bedrooms, servant room, number of balconies, number of houses of years a house has been built – predict the price of a house.
14. Implement a classification/logistic regression problem. For example, based on different features of students data, classify whether a student is suitable for a particular activity. Based on the available dataset, a student can also implement another classification problem like checking whether an email is spam or not.

15. Use some function for regularization of dataset based on problem 14.

16. Use some function for neural networks, like Stochastic Gradient Descent or back propagation algorithm to predict the value of a variable based on the dataset of problem
Instruction Hours / week: L: 0  T: 0  P: 4  Marks: Int : 40  Ext : 60  Total: 100

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 a multiplication table for numbers up to 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 a string is a palindrome.
7. Implement linear search.
8. Implement binary search.
9. Implement matrices addition, subtraction and multiplication
10. Fifteen students were enrolled in a course. There ages were: 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 for 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 a frequency distribution of systolic blood pressure. Compute all the measures of dispersion.

<table>
<thead>
<tr>
<th>Midpoint</th>
<th>95.5</th>
<th>105.5</th>
<th>115.5</th>
<th>125.5</th>
<th>135.5</th>
<th>145.5</th>
<th>155.5</th>
<th>165.5</th>
<th>175.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>5</td>
<td>8</td>
<td>22</td>
<td>27</td>
<td>17</td>
<td>9</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

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 aCSUount. Also, create a repo using Github.
17CSU514A  Oracle (SQL/PL-SQL) LAB - PRACTICAL  Semester – V

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

[SQL COMMANDS]

1) SQL* formatting commands
2) To create a table, alter and drop table.
3) To perform select, update, insert and delete operation in a table.
4) To make use of different clauses viz where, group by, having, order by, union and intersection,
5) To study different constraints.

[SQL FUNCTION]

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

[PL/SQL]

11) To understand working with PL/SQL
12) To implement Cursor on a table.
13) To implement trigger on a table
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. WAP to display the first n terms of Fibonacci series.
5. WAP to find factorial of the given number.
6. WAP to find sum of the following series for n terms: 1 – 2/2! + 3/3! - - - - - n/n!
7. WAP to calculate the sum and product of two compatible matrices.
SCOPE
The main objective of the course is to portray the recent trends in the field of cloud computing and providing exposures to some open source and commercial clouds.

OBJECTIVES
- Provide a good understanding of the concepts, standards and protocols in Cloud computing

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

WEB SITES
1. en.wikipedia.org/wiki/Cloud_computing
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.
- To train students in developing skills for writing system software with the aid of sophisticated OS services, programming languages and utility tools.

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, yaCSU. 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
## SCOPE
This course introduces students to the basic concepts and techniques of Data Mining, develop skills of using recent data mining software for solving practical problems, gain experience of doing independent study and research.

### OBJECTIVES
- To introduce students to the basic concepts and techniques of Data Mining.
- To develop skills of using recent data mining software for solving practical problems.
- To gain experience of doing independent study and research.
- Possess some knowledge of the concepts and terminology associated with database systems, statistics, and machine learning.

### 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
WEB SITES
1. Thedacs.Com
2. Dwreview.Com
3. Pcai.Com
4. Eruditionhome.Com
SCOPE
This course presents an introduction to computer graphics designed to give the student an overview of fundamental principles. The course will include an overview of common graphics hardware, 2D and 3D transformations and viewing, and basic raster graphics concepts such as scan-conversion, and clipping. Methods for modeling objects as polygonal meshes or smooth surfaces, and as rendering such as hidden-surface removal, shading, illumination, and shadows will be investigated.

OBJECTIVES
- Have a knowledge and understanding of the structure of an interactive computer graphics system, and the separation of system components.
- Have a knowledge and understanding of geometrical transformations and 3D viewing.
- Be able to create interactive graphics applications.
- Have a knowledge and understanding of techniques for representing 3D geometrical objects.
- Have a knowledge and understanding of the fundamental principles of local and global illumination models.

UNIT-I

UNIT-II

UNIT-III
Fundamental Techniques in Graphics : Raster scan line, circle and ellipse drawing, thick primitives, Polygon filling, line and polygon clipping algorithms, 2D and 3D Geometric Transformations, 2D and 3D Viewing Transformations (Projections- Parallel and Perspective), Vanishing points.

UNIT-IV
Geometric Modeling : Representing curves & Surfaces.

UNIT V

Suggested Readings


17CSU603A  PHP PROGRAMMING  3H – 3C

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

SCOPE
This course design focuses on the structure of the website including the information architecture, the layout or the pages and the conceptual design with branding. PHP helps the students for developing dynamic web pages.

OBJECTIVES
- To work with open source applications that deal with database and website development.
- Establish a working environment for PHP web page development
- Very familiar with GUI, coded, modified controls

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
5. David Sklar, & Adam Trachtenberg. 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

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

2. Create virtual machines that access different programs on different platforms.

3. Working on tools used in cloud computing online-
   a) Storage
   b) Sharing of data
   c) Manage your calendar, to-do lists,
   d) A document editing tool

4. Exploring Google cloud

5. Exploring Microsoft cloud

6. Exploring Amazon cloud
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.
Practical exercises based on concepts listed in theory.
1. Write a program to implement Bresenham’s line drawing algorithm.

2. Write a program to implement mid-point circle drawing algorithm.

3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.

4. Write a program to clip a polygon using Sutherland Hodgeman algorithm.

5. Write a program to apply various 2D transformations on a 2D object (use homogenous coordinates).

6. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.

7. Write a program to draw Hermite/Bezier curve.
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 accepts 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.
1. Write a shell script to check if the number entered at the command line is prime or not.
2. Write a shell script to modify —call command to display calendars of the specified months.
3. Write a shell script to modify —call command to display calendars of the specified range of months.
4. Write a shell script to accept a login name. If not a valid login name display message — Entered login name is invalid.
5. Write a shell script to display date in the mm/dd/yy format.
6. Write a shell script to display on the screen sorted output of —who command along with the total number of users.
7. Write a shell script to display the multiplication table any number,
8. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
9. Write a shell script to find the sum of digits of a given number.
10. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
11. Write a shell script to find the LCD (least common divisor) of two numbers.
12. Write a shell script to perform the tasks of basic calculator.
13. Write a shell script to find the power of a given number.
14. Write a shell script to find the binomial coefficient C(n, x).
15. Write a shell script to find the permutation P(n,x).
16. Write a shell script to find the greatest number among the three numbers.
17. Write a shell script to find the factorial of a given number.
18. Write a shell script to check whether the number is Armstrong or not.
19. Write a shell script to check whether the file have all the permissions or not.
Semester – VI

17CSU604    BIG DATA ANALYTICS     4H- 4C

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

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 big data.

OBJECTIVES: To impart to students the skills required to design scalable systems that can accept, store, and analyze large volumes of unstructured 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:


WEB SITES

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