

FAKIR MOHAN UNIVERSITY

B. Sc (Computer Science)

SEMESTER-I, COMP SC.(H)-CC-I

(TH.): PROGRAMMING USING “C”

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

UNIT- I

Introduction to Programming Language, Introduction to C Programming , Character Set, C Tokens, Keywords & Identifiers, Constants, Variables, Data Types, Variables , Storage Classes, Operators (Arithmetic, Relational, Logical , Assignment, Increment & Decrement, Conditional, Bitwise), Expressions , Input and Output Operations.

UNIT- II

Decision Making and Branching: Simple IF Statement, IF ELSE Statement, Nesting IF ELSE Statement, ELSE IF Ladder, Switch Statement, ?Operator, GOTO Statement. Decision Making and Looping: The WHILE Statement, The DO Statement, The FOR Statement, Jumps in LOOPS. Arrays, Character Arrays and Strings.

UNIT- III

User-defined Functions: Need, Elements & Definition, Function Calls, Function Definition, Category of Functions, Recursion. Structures and Unions: Defining, Declaring, Accessing, Initialization Structure, Arrays of Structures, Arrays within Structures, Structures and Functions, Unions.

UNIT- IV

Pointers: Accessing the Address of a Variable, Declaring Pointer Variables, Initializations of Pointer Variable, Accessing a Variable through its Pointer, Chain of Pointers, Pointer Expressions, Pointer Increments and Scale Factor, Pointers and Arrays, Pointers and Character Strings, Array of Pointers, Pointers as Function Arguments, Functions Returning Pointers, Pointers to Functions, Pointers to Structures, Troubles with Pointers.

UNIT- V

File Management in C: Defining and Opening a File, Closing a File, Input/ Output Operations on Files, Error Handling during I/O Operations, Random Access to Files, Command Line Arguments, Dynamic Memory Allocation.

Text Book:

Programming in ANSI C: E. Balguruswamy4/e (TMH)

SEM-I, COMP. SC.(H)-CC-II (TH.) - COMPUTER ORGANIZATION

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

UNIT-I

Character Codes, Decimal System, Binary System, Decimal to Binary Conversion, Hexadecimal Notation, Boolean Algebra, Basic Logic Functions: Electronic Logic Gates, Synthesis of Logic Functions, Minimization of Logic Expressions, Minimization using Karnaugh Maps, Synthesis with NAND and NOR Gates.

UNIT-II

Flip-Flops, Gated Latches, Master-Slave Flip-Flops, Edge-Triggering, T Flip-Flops, JK Flip-Flops. Registers and Shift Registers, Counters, Decoders, Multiplexers, Programmable Logic Devices (PLDs), Programmable Array Logic (PAL), Complex Programmable Logic Devices (CPLDs), Field-Programmable Gate Array (FPGA), Sequential Circuits, Timing Diagrams, The Finite State Machine Model, Synthesis of Finite State Machines.

UNIT-III

Basic Structure of Computers: Computer Types, Functional Units, Input Unit, Memory Unit, Arithmetic and Logic Unit, Output Unit, Control Unit, Basic Operational Concepts, Bus Structures, Software. Machine Instructions and Programs: Numbers, Arithmetic Operations and Characters: Number Representation, Addition of Positive Numbers, Addition and Subtraction of Signed Numbers, Overflow of Integer Arithmetic, Characters, Memory Locations and Addresses, Byte Addressability, Word Alignment, Accessing Numbers, Characters and Character Strings, Memory Operations, Instructions and Instruction Sequencing, Register Transfer Notation, Basic Instruction Types, Instruction Execution and Straight-Line Sequencing, Branching, Condition Codes, Generating Memory Addresses, Addressing Modes, Implementation of Variables and Constants, Indirection and Pointers, Indexing and Arrays, Relative Addressing.

UNIT-IV

THE ARM EXAMPLE: Registers, Memory Access, and Data Transfer, Register Structure, Memory Access Instructions and Addressing Modes, Register Move Instructions, Arithmetic and Logic Instructions: Arithmetic Instructions, Logic Instructions, Branch Instructions, Setting Condition Codes, Assembly Language, Pseudo-Instructions, I/O Operations, Subroutines, Vector Dot Product Program, Byte-Sorting Program, Linked-List Insertion and Deletion Subroutines. Basic Input-Output Operations, Stacks and Queues, Subroutines. PowerPC Example: Basic PowerPC Processor Organization, Load and Store Instructions, Arithmetic and Logic Instructions, Flow Control Instructions, Compare Instructions, Logic Instructions, Subroutines.

UNIT-V

Memory System: Semiconductor RAM Memories, Internal Organization of Memory Chips, Static Memories, Asynchronous DRAMS, Synchronous DRAMS, Structure of Large Memories, Memory System Considerations, RAMBUS Memory. Read-Only Memories: ROM, PROM, EPROM, EEPROM, Flash Memory, Speed, Size, and Cost of Memory. Secondary Storage: Magnetic Hard Disks, Optical Disks, Magnetic Tape Systems.

Text Book:

Carl Hamacher, Z. Vranesic, S. Zaky: Computer Organization, 5/e (TMH)

Reference Book:

William Stallings: Computer Organization and Architecture (Design for Performance), 9/e

BOTH COMP SC.- GE-I, & DSC-1/2/3-I, ARE SAME AS COMP. SC. CC-I, SEM-I

B. Sc. (Computer Science)
CC-I (Programming Using C)
List of Programs for Laboratory Work

1. Write a program in C programming language to illustrate the basic arithmetic operations.
2. Write a program in C programming language to illustrate the usage of ++ (post and pre-increment) and -- (post and pre-decrement) operators, global and internal variables, bitwise operator, and size of data type.
3. Write a program in C programming language to create the pyramid pattern.
4. Write a program in C programming language to add two very large integer numbers (at least 100 digits in each).
5. Write a program in C programming language to find roots of quadratic equation.
6. Write a program in C programming language to find the area of a circle.
7. Write a program in C programming language to find mean, variance, and standard deviation.
8. Write a program in C programming language to find the area of isosceles triangle.
9. Write a program in C programming language to convert lowercase to uppercase and vice-versa.
10. Write a program in C programming language to concatenation of two strings.
11. Write a program in C programming language to check whether a given string is palindrome or not.
12. Write a program in C programming language to count only the number of characters in a document.
13. Write a program in C programming language to illustrate the idea of call by value and call by address through swapping of two numbers and other relevant examples.
14. Write a program in C programming language to illustrate the idea of recursion through factorial of a number, Fibonacci numbers, GCD, and LCD.
15. Write a program in C programming language to illustrate the use of single dimensional array and multi-dimensional array by giving examples like bubble sort, matrix addition, matrix multiplication, inverse of a matrix, determinant of a matrix, etc.
16. Write a program in C programming language to illustrate the use of structure and union.
17. Write a program in C programming language to store a value in file and retrieve from file.
18. Write a program in C programming language to test whether a number is Armstrong or not.
19. Write a program in C programming language to find all prime numbers from a certain range like 1-10000.
20. Write a program in C programming language to cyclically permute the array.

SEM – II, COMP. SC.(H) CC – III - PROGRAMMING USING C++

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

UNIT- I

Principles of Object-Oriented Programming: Object-Oriented Programming (OOP) Paradigm, Basic Concepts of OOP, Benefits of OOP, Object Oriented Languages, Applications of OOP. Beginning with C++: Applications of C++, C++ statements, Example with Class, Structure of C++ Program, Creating the Source File, Compiling and Linking. Tokens, Expressions and Control Structures: Tokens, Keywords, Identifiers & Constants, Basic Data Types, User-Defined Data Types, Derived Data Types, Symbolic Constants, Type Compatibility, Declaration of Variables, Dynamic Initialization of Variables, Reference Variables, Operators in C++, Scope Resolution Operator, Member Differencing Operators, Memory Management Operators, Manipulators, Type Cast Operators, Expressions and their Types, Special Assignment Expressions, Implicit Conversions, Operator Overloading, Operator Precedence, Control Structure

UNIT- II

Functions in C++: The Main Function, Function Prototyping, Call By Reference, Return by Reference, Inline Functions, Default Arguments, Const. Arguments, Function Overloading, Friend & Virtual Functions, Math. Library Functions. Classes and Objects: Specifying a Class, Defining Member Functions, Making an outside Function Inline, Nested Member Functions, Private Member Functions, Arrays within a Class, Memory Allocation for Objects, Static Data Members, Static Member Functions, Arrays of Objects, Objects as Function Arguments, Friendly Functions, Returning Objects, Cons. Member Functions, Pointer to Members, Local Classes.

UNIT- III

Constructors & Destructors: Constructors, Parameterized Constructors, Multiple Constructors in a Class, Constructors with Default Arguments, Dynamic Initialization of Objects, Copy Constructor, Dynamic Constructors, Constructing Two-Dimensional Arrays, Const. Objects, Destructors. Operator Overloading and Type Conversions: Defining Operator Overloading, Overloading Unary Operators, Overloading Binary Operators, Overloading Binary Operators using Friends, Manipulation of Strings using Operators, Rules for Overloading Operators, Type Conversions.

UNIT- IV

Inheritance: Defining Derived Classes, Single Inheritance, Making a Private Member Inheritance, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Hybrid Inheritance, Virtual Base Classes, Abstract Classes, Constructors in Derived Classes, Member Classes, Nesting of Classes. Pointers, Virtual Functions and Polymorphism: Pointers, Pointers to Objects, this Pointer, Pointers to Derived Classes, Virtual Functions, Pure Virtual Functions.

UNIT- V

Managing Console I/O Operations: C++ Streams, C++ Stream Classes, Unformatted I/O Operations, Formatted Console I/O Operations, Managing Output with Manipulators. Files: Classes for File Stream Operations, Opening and Closing a File, Detecting end-of-file, File Modes, File Pointers and their Manipulations, Sequential Input and Output Operations, Updating a File: Random Access, Error Handling During File Operations, Command-line Arguments.

Text Book:

Object Oriented Programming with C++: E. Balgurusamy, 4/e (TMH).

SEM – II, COMP. SC.(H) CC – IV - DATA STRUCTURE

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

UNIT-I

Introduction and Overview: Definitions, Concept of Data Structures, Overview of Data Structures, Implementation of Data Structures. Arrays: Terminology, One-Dimensional Array, Multi-Dimensional Arrays, Pointer Arrays.

UNIT-II

Linked Lists: Single Linked List, Circular Linked List, Double Linked List, Circular Double Linked List, Application of Linked Lists, Memory Representation, Boundary Tag System, De-allocation Strategy, Buddy System, Compaction.

UNIT-III

Stacks: Definition, Representation of Stack (Array, Linked List), Operations on Stacks, Applications of Stack (Evaluation of Arithmetic Expressions, Code Generation, Implementation of Recursion, Factorial Calculation, Quick Sort, Tower of Hanoi, Activation Record Management).

UNIT – IV

Queues: Definition, Representation of Queues (Array, Linked List), Circular Queue, Deque, Priority Queue, Application of Queues (Simulation, CPU Scheduling in Multiprogramming Environment, Round Robin Algorithm).

UNIT – V

Tree: Binary Trees, Properties of Binary Tree, Linear Representation of Binary a Binary Tree, Linked Representation of a Binary Tree, Physical Implementation of Binary Tree in Memory, Operations on Binary Tree (Insertion, Deletion, Traversal, Merging of two Binary Trees), Types of Binary Trees (Expression Tree, Binary Search Tree, Heap Tree, Threaded Binary Trees, Height Balanced Binary Tree, Weighted Binary Tree, Decision Trees).

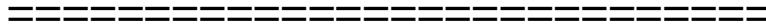
TEXT BOOK:

Classic Data Structures: D. SAMANTA (PHI).

**COMP. SC.-GE – II IS SAME AS CC – IV COMP.
SC.-DSC-1/2/3- II IS SAME AS CC – II**

Programming Using C++

CC-IV



VARIABLE, OPERATOR AND EXPRESSION

- Question 1** Write a program to print HELLO WORLD on screen.
- Question 2** Write a program to display the following output using a single cout statement.
- | Subject | Marks |
|-------------|-------|
| Mathematics | 90 |
| Computer | 77 |
| Chemistry | 69 |
- Question 3** Write a program which accept two numbers and print their sum.
- Question 4** Write a program which accept temperature in Fahrenheit and print it in centigrade.
- Question 5** Write a program which accept principle, rate and time from user and print the simple interest.
- Question 6** Write a program which accepts a character and display its ASCII value.
- Question 7** Write a program to swap the values of two variables.
- Question 8** Write a program to calculate area of circle.
- Question 9** Write a program to check whether the given number is positive or negative (using ? : ternary operator)
- Question 10** Write a program to check whether the given number is even or odd (using ? : ternary operator)

FLOW OF CONTROL

- Question 1** Any integer is input by the user. Write a program to find out whether it is an odd number or even number.
- Question 2** Find the absolute value of a number entered by the user.
- Question 3** Write a program to calculate the total expenses. Quantity and price per item are input by the user and discount of 10% is offered if the expense is more than 5000.
- Question 4** Write a program to determine whether the seller has made profit or incurred loss. Also determine how much profit he made or loss he incurred. Cost price

and selling price of an item is input by the user.

- Question 5** If the ages of Ram, Sulabh and Ajay are input by the user, write a program to determine the youngest of the three.
- Question 6** Write a program to check whether a triangle is valid or not, when the three angles of the triangle are entered by the user. A triangle is valid if the sum of all the three angles is equal to 180 degrees.
- Question 7** Any year is input by the user. Write a program to determine whether the year is a leap year or not.
- Question 8** In a company an employee is paid as under:
If his basic salary is less than Rs. 1500, then HRA = 10% of basic salary and DA = 90% of basic salary.
If his salary is either equal to or above Rs. 1500, then HRA = Rs. 500 and DA = 98% of basic salary.
If the employee's salary is input by the user write a program to find his gross salary.
- Question 9** Write a program to calculate the monthly telephone bills as per the following rule:
Minimum Rs. 200 for upto 100 calls.
Plus Rs. 0.60 per call for next 50 calls.
Plus Rs. 0.50 per call for next 50 calls.
Plus Rs. 0.40 per call for any call beyond 200 calls.
- Question 10** Write a program to find the roots of and quadratic equation of type ax^2+bx+c where a is not equal to zero.
- Question 11** The marks obtained by a student in 5 different subjects are input by the user. The student gets a division as per the following rules:
Percentage above or equal to 60 - First division
Percentage between 50 and 59 - Second division
Percentage between 40 and 49 - Third division
Percentage less than 40 - Fail
Write a program to calculate the division obtained by the student.
- Question 12** Any character is entered by the user; write a program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol. The following table shows the range of ASCII values for various characters.

Characters	ASCII Values
A – Z	65 – 90
a – z	97 – 122
0 – 9	48 – 57
special symbols	0 - 47, 58 - 64, 91 - 96, 123 – 127

CLASS AND OBJECT

Question 1 Define a class student with the following specification

Private members of class student

admno	integer
sname	20 character
eng, math, science	float
total	float
ctotal()	a function to calculate eng + math + science with float return type.

Public member function of class student

Takedata()	Function to accept values for admno, sname, eng, science and invoke ctotal() to calculate total.
Showdata()	Function to display all the data members on the screen.

Question 2 Define a class batsman with the following specifications:

Private members:

bcode	4 digits code number
bname	20 characters
innings, notout, runs	integer type
batavg	it is calculated according to the formula $batavg = runs / (innings - notout)$
calcavg()	Function to compute batavg

Public members:

readdata()	Function to accept value from bcode, name, innings, notout and invoke the function calcavg()
displaydata()	Function to display the data members on the screen.

Question 3 Define a class TEST in C++ with following description:

Private Members

TestCode of type integer
 Description of type string
 NoCandidate of type integer
 CenterReqd (number of centers required) of type integer
 A member function CALCNTR() to calculate and return the number of centers as $(NoCandidates/100+1)$

Public Members

- A function SCHEDULE() to allow user to enter values for TestCode, Description, NoCandidate & call function CALCNTR() to calculate the number of Centres
- A function DISPTEST() to allow user to view the content of all the data members

Question 4 Define a class in C++ with following description:

Private Members

A data member Flight number of type integer
 A data member Destination of type string
 A data member Distance of type float
 A data member Fuel of type float
 A member function CALFUEL() to calculate the value of Fuel as per the following criteria

Distance	Fuel
≤ 1000	500

more than 1000 and <=2000 1100
more than 2000 2200

Public Members

A function FEEDINFO() to allow user to enter values for Flight Number, Destination, Distance & call function CALFUEL() to calculate the quantity of Fuel

A function SHOWINFO() to allow user to view the content of all the data members

Question 5 Define a class BOOK with the following specifications :

Private members of the class BOOK are

BOOK NO integer type
BOOKTITLE 20 characters
PRICE float (price per copy)
TOTAL_COST() A function to calculate the total cost for N number of copies where N is passed to the function as argument.

Public members of the class BOOK are

INPUT() function to read BOOK_NO, BOOKTITLE, PRICE
PURCHASE() function to ask the user to input the number of copies to be purchased. It invokes TOTAL_COST() and prints the total cost to be paid by the user.

Note : You are also required to give detailed function definitions.

Question 6 Define a class REPORT with the following specification:

Private members :

adno 4 digit admission number
name 20 characters
marks an array of 5 floating point values
average average marks obtained
GETAVG() a function to compute the average obtained in five subject

Public members:

READINFO() function to accept values for adno, name, marks. Invoke the function GETAVG()
DISPLAYINFO() function to display all data members of report on the screen.

You should give function definitions.

CONSTRUCTOR AND DESTRUCTOR

Question 1 Answer the questions (i) and (iii) after going through the following

class:

```
class Seminar
{
    int time;
public:
    Seminar()          //Function 1
    {
        time = 30;
        cout << "Seminar starts now" << endl;
    }

    void lecture()     //Function 2
    {
        cout << "Lectures in the seminar on" << endl;
    }

    Seminar(int duration) //Function 3
    {
        time = duration;
        cout << "Seminar starts now" << endl;
    }

    ~Seminar()        //Function 4
    {
        cout << "Thanks" << endl;
    }
};
```

- i. Write statements in C++ that would execute Function 1 and Function 3 of class Seminar.
- ii. In Object Oriented Programming, what is Function 4 referred as and when does it get invoked/called?
- iii. In Object Oriented Programming, which concept is illustrated by Function 1 and Function 3 together?

Question 2 Answer the questions (i) and (ii) after going through the following class:

```
class Test
{
    char paper[20];
    int marks;
public:
    Test ()          // Function 1
    {
        strcpy (paper, "Computer");
        marks = 0;
    }
};
```

```

Test (char p[])    // Function 2
{
    strcpy(paper, p);
    marks = 0;
}

Test (int m)      // Function 3
{
    strcpy(paper, "Computer");
    marks = m;
}

Test (char p[], int m)    // Function 4
{
    strcpy (paper, p);
    marks = m;
}
};

```

- i. Write statements in C++ that would execute Function 1, Function 2, Function 3 and Function 4 of class Test.
- ii. Which feature of Object Oriented Programming is demonstrated using Function 1, Function 2, Function 3 and Function 4 together in the above class Test?

3

Consider the definition of the following class:

```

class Sample
{
private:
    int x;
    double y;
public :
    Sample(); //Constructor 1
    Sample(int); //Constructor 2
    Sample(int, int); //Constructor 3
    Sample(int, double); //Constructor 4
};

```

- i. Write the definition of the constructor 1 so that the private member variables are initialized to 0.
- ii. Write the definition of the constructor 2 so that the private member variable x is initialized according to the value of the parameter, and the private member variable y is initialized to 0.
- iii. Write the definition of the constructors 3 and 4 so that the

private
member variables are initialized according to the values of the
parameters.

INHERITANCE

Question 1 Consider the following declaration and answer the questions given below :

```
class PPP
{
    int H;
    protected :
        int S;
    public :
        void INPUT (int);
        void OUT();
};
class QQQ : private PPP
{
    int T;
    protected :
        int U;
    public :
        void INDATA(int, int);
        void OUTDATA();
};
class RRR : public QQQ
{
    int M;
    public :
        void DISP( void );
};
```

- (i) Name the base class and derived class of the class QQQ.
- (ii) Name the data member(s) that can be accessed from function DISP().
- (iii) Name the member function(s), which can be accessed from the objects of class RRR.
- (iv) Is the member function OUT() accessible by the object of the class QQQ?

Question 2 Answer the questions (i) to (iv) based on the following:

```
class PUBLISHER
{
    char Pub[12];
    double Turnover;
protected:
    void Register();
public:
```

```

        PUBLISHER();
        void Enter();
        void Display();
};
class BRANCH
{
    char CITY[20];
protected:
    float Employees;
public:
    BRANCH();
    void Haveit();
    void Giveit();
};

class AUTHOR:private BRANCH,public PUBLISHER
{
    int Acode;
    char Aname[20];
    float Amount;
public:
    AUTHOR();
    void Start();
    void Show();
};

```

- i) Write the names of data members, which are accessible from objects belonging to class AUTHOR.
- ii) Write the names of all the member functions which are accessible from objects belonging to class BRANCH.
- iii) Write the names of all the members which are accessible from member functions of class AUTHOR.
- iv) How many bytes will be required by an object belonging to class AUTHOR?

Question 3 Consider the following declarations and answer the question given below :

```

class vehicle
{
private:
    int wheels;
protected :
    int passenger;
public :
    void inputdata (int, int);
    void outputdata();
};
class heavyvehicle : protected vehicle
{
    int diesel_petrol;
protected :
    int load;
};

```

```

public:
    void readdata(int, int);
    void writedata();
};
class bus : private heavyvehicle
{
    char make[20];
public :
    void fetchdata(char);
    void displaydata();
};

```

- (i) Name the base class and derived class of the class heavy_vehicle.
- (ii) Name the data member(s) that can be accessed from function displaydataO.
- (iii) Name the data member's that can be accessed by an object of bus class.
- (iv) Is the member function outputdata() accessible to the objects of heavy_vehicle class.

Question 4 Answer the questions (i) to (iv) based on the following code :

```

class Drug
{
    char Category[10];
    char Date_of_manufacture[10];
    char Company[20];
public:
    Drug();
    void enterdrugdetails();
    void showdrugdetails{};
};
class Tablet : public Drug
{
    protected:
    char tablet_name[30];
    char Volume_label[20];
public:
    float Price;
    Tablet();
    void entertabletdetails();
    void showtabletdetails ();
};
class PainReliever : public Tablet
{
    int Dosage_units;
    char Side_effects[20];
    int Use_within_days;
public:
    PainReliever();
    void enterdetails();
    void showdetails();
};

```

- (i) How many bytes will be required by an object of class Drug and an object of class

PainReliever respectively ?

- (ii) Write names of all the data members which are accessible from the object of class PainReliever.
- (iii) Write names of all the members accessible from member functions of class Tablet.
- (iv) Write names of all the member functions which are accessible from objects of class PainReliever.

FILE HANDLING - TEXT FILE

Question 1 Write a C++ program to write number 1 to 100 in a data file NOTES.TXT.

Question 2 Write a C++ program, which initializes a string variable to the content "Time is a great teacher but unfortunately it kills all its pupils. Berlioz" and outputs the string to the disk file OUT.TXT. you have to include all the header files if required.

Question 3 Write a user-defined function in C++ to read the content from a text file OUT.TXT, count and display the number of alphabets present in it.

Question 4 Write a function to count the number of blank present in a text file named "OUT.TXT".

Question 5 Write a function to count number of words in a text file named "OUT.TXT".

Question 6 Write a function in C++ to print the count of word the as an independent word in a text file STORY.TXT.

for example, if the content of the file STORY.TXT is
There was a monkey in the zoo. The monkey was very naughty.

Then the output of the program should be 2.

Question 7 Write a function in C++ to count and display the number of lines not starting with alphabet 'A' present in a text file "STORY.TXT".

Example:

If the file "STORY.TXT" contains the following lines,

The rose is red.

A girl is playing there.

There is a playground.

An aeroplane is in the sky.

Numbers are not allowed in the password.

The function should display the output as 3.

Question 8 Assuming that a text file named FIRST.TXT contains some text written into it, write a function named copyupper(), that reads the file FIRST.TXT and creates a new file named SECOND.TXT contains all words from the file FIRST.TXT in uppercase.

Question 9 Assuming that a text file named FIRST.TXT contains some text written into it, write a function named vowelwords(), that reads the file FIRST.TXT and creates a new file named SECOND.TXT, to contain only those words from the file FIRST.TXT which start with a lowercase vowel (i.e., with 'a', 'e', 'i', 'o', 'u').

For example, if the file FIRST.TXT contains

Carry umbrella and overcoat when it rains
Then the file SECOND.TXT shall contain
umbrella and overcoat it

DATA STRUCTURE - STACK AND QUEUE

- Question 1** Complete the class with all function definitions for a stack
- ```
class stack
{
 int data[10];
 int top;
public :
 stack(){top=-1;}
 void push();
 void pop();
}
```
- Question 2** Change the following infix expression postfix expression.  
(A + B)\*C+D/E-F
- Question 3** Convert the expression (True && False) || !(False || True) to postfix expression.  
Show the contents of the stack at every step.
- Question 4** Use a stack to evaluate the following postfix expression and show the content of the stack after execution of each operation. Don't write any code. Assume as if you are using push and pop member functions of the stack.  
AB-CD+E\*+ (where A=5, B=3, C=5, D =4, and E=2)
- Question 5** Evaluate the following postfix expression using a stack and show the contents of stack after execution of each operation :  
50,40,+,18, 14,-, \*,+
- Question 6** Evaluate the following postfix expression using a stack and show the contents of stack after execution of each operation :  
TRUE, FALSE, TRUE, FALSE, NOT, OR, TRUE, OR, OR, AND
- Question 7** Complete the class with all function definitions for a circular queue
- ```
class queue
{
    int data[10];
    int front, rear;
public :
    queue(){front=-1;rear=-1 }
```



```

        void add();
        void remove();
    }

```

Question 8 Each node of a STACK contains the following information, in addition to required pointer field :

- i) Roll number of the student
- ii) Age of the student

Give the structure of node for the linked stack in question TOP is a pointer which points to the topmost node of the STACK. Write the following functions.

- i) PUSH() - To push a node to the stack which is allocated dynamically
- ii) POP() - To remove a node from the stack and release the memory.

Question 9 Write a function in C++ to perform a DELETE operation in a dynamically allocated queue considering the following description :

```

struct Node
{
    float U,V;
    Node *Link;
};
class QUEUE
{
    Node *Rear,*Front;
public:
    QUEUE(){Rear=NULL; Front=NULL;}
    void INSERT();
    void DELETE();
    ~QUEUE();
};

```

Question 10 Give the necessary declaration of a linked list implemented queue containing float type values. Also write a user-defined function in C++ to delete a float type number from the queue.

B. Sc. (Computer Science)
 GE-II = CC-IV (Data Structure)
 List of Programs for Laboratory Work

1. Write a program in high level programming language to illustrate the use of Array –one dimensional, multi-dimensional and pointer arrays.
2. Write a program in high level programming language to illustrate the operations like insertion, deletion in both sides of singly linked list.
3. Write a program in high level programming language to illustrate the use of circular linked list, Double linked list, and circular double linked list.
4. Write a program in high level programming language to illustrate the idea of compaction.

5. Write a program in high level programming language to illustrate the properties of stack (FILO) in both array and linked list implementation.
6. Write a program in high level programming language to illustrate the use of stack data structure in factorial calculation, evaluation of arithmetic operations, quick sort, and tower of Hanoi.
7. Write a program in high level programming language to illustrate the operations of queue in both array and linked list implementations.
8. Write a program in high level programming language to find the usefulness of priority queue.
9. Write a program in high level programming language to illustrate the use of queue in round-robin scheduling algorithm.
10. Write a program in high level programming language to illustrate the use of Queue and Stack in BFS and DFS respectively.
11. Write a program in high level programming language to implement binary tree in array and linked list.
12. Write a program in high level programming language to implement binary tree traversal.
13. Write a program in high level programming language to illustrate the use of binary search tree, threaded binary tree, and height balanced tree.
14. Write a program in high level programming language to illustrate the use of weighted binary tree.

Write a program in high level programming language to illustrate the use of decision tree

CHOICE BASED CREDIT SYSTEM SYLLABUS

Core Courses Semester-III COMP. SC. (Hons)

CC- V - OPERATING SYSTEMS

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

UNIT- I

Operating System, Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security, Distributed Systems, Special Purpose Systems, Computing Environments, Open-Source Operating Systems. Operating System Services, User Operating System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and

Implementation, Operating System Structure, Virtual Machines, Operating System Debugging, Operating System Generations, System Boot.

UNIT- II

Process: Process Concept, Process Scheduling, Operations on Processes, Inter-Process Communication, Examples of IPC Systems, Communication in Client-Server Systems. Multithreaded Programming: Multithreading Models, Thread Libraries, Threading Issues, Operating-System Examples.

UNIT- III

Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling. Multiple-Process Scheduling. Synchronization: The Critical Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Monitors, Synchronization Examples, Atomic Transactions.

UNIT- IV

Deadlocks: System Model, Deadlock Characterization, Methods of Handling Deadlocks, Deadlock Prevention, Deadlock avoidance, Deadlock Detection, Recovery from Deadlock. Memory Management Strategies: Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation, Example: The Intel Pentium.

UNIT- V

Virtual-Memory Management: Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files, Allocating Kernel Memory. File System: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing, Protection.

TEXT BOOK:

Operating System Concepts: Silberschatz, Galvin, Gagne, 8/e (Wiley-India)

Semester-III COMP. SC. (Hons)

CC- VI - DATABASE MANAGEMENT SYSTEM

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

UNIT-I

Databases and Database Users, Database System Concepts and Architecture, Data Modelling using the Entity-Relationship(ER) Model, The Enhanced Entity-Relationship (EER) Model.

UNIT-II

Relational Model: The Relational Data Model and Relational Database Constraints, The Relational Algebra and Relational Calculus.

UNIT-III

Relational Database Design by ER- and EER-to-Relational Mapping, SQL-99: Schema Definition, Constraints, Queries, and Views, Introduction to SQL Programming Techniques.

UNIT-IV

Functional Dependencies and Normalization for Relational Databases, Relational Database Algorithms and Further Dependencies, Practical Database Design Methodology and use of UML Diagrams.

UNIT-V

Disk Storage, Basic File Structures, and Hashing, Indexing Structures for Files, Algorithms for Query Processing and Optimization, Physical Database Design and Tuning.

TEXT BOOK:

Fundamentals of Database Systems, RamezElmasri&Shamkant B. Navathe, Pearson (Fifth Edition)

Semester-III COMP. SC. (Hons)

CC- VII - DISCRETE MATHEMATICAL STRUCTURES

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

UNIT-I

Logic and Proofs: Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Normal Forms, Proof Methods and Strategy, Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms.

UNIT-II

Basic Structures: Sets, Set Operations, Functions, Recursive Functions, Sequences and Summations. Relations: Relations and their Properties, n-ary Relations and their Applications, Representing Relations, Closures of Relations, Equivalence Relations, Partial Ordering. Boolean

UNIT-III

Algebra: Boolean Functions, Representing Boolean Functions, Logic Gates, Minimization of Circuits. Algebraic Structures & Coding Theory: The Structure of Algebras, Semi-groups,

Monoids and Groups, Homomorphism, Normal Subgroups, and Congruence Relations, Rings, Integral Domains and Fields, Quotient and Product Algebras, Coding Theory. Polynomial Rings and Polynomial Codes.

UNIT-IV

Counting: Basics of Counting, The Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients, Generalized Permutations and Combinations, Generating Permutations and Combinations. Advanced Counting Techniques, Applications of Inclusion-Exclusion, Discrete probability, Conditional probability, Bayes' Theorem.

UNIT-V

Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Havel-Hakimi Theorem, Representing Graphs and Graph Isomorphism, Connectivity, Cut-Sets, Euler and Hamiltonian Paths, Shortest-Path Problem, Planar Graphs, Graph Coloring, Network Flows.

Text Book:

Kenneth H Rosen, Discrete Mathematics & Its Applications, McGraw-Hill. 7/e

Semester-IV COMP. SC. (Hons)

CC-VIII - JAVA PROGRAMMING

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

UNIT-I

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

UNIT-II

Arrays, Strings and I/O: Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files. Object-Oriented Programming Overview: Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

UNIT-III

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

UNIT-IV

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

UNIT-V

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

Text Book:

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

Semester-IV COMP. SC. (Hons)

CC-IX - COMPUTER NETWORK

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

UNIT-I

Introduction: Data Communications, Networks, The Internet, Protocols and Standards. Network Models: Layered Tasks, The OSI Model, Layers in the OSI Model, TCP/ IP Protocol Suite, Addressing.

UNIT-II

Data and Signals: Analog and Digital, Periodic Analog Signals, Digital Signals, Transmission Impairment, Data Rate Limits, Performance. Digital Transmission: Digital-To-Digital Conversion, Analog-To-Digital Conversion, Transmission Modes. Analog Transmission: Digital-To-Analog Conversion, Analog-To-Analog Conversion.

UNIT-III

Multiplexing and Spreading: Multiplexing, Spread Spectrum. Transmission Media: Guided Media, Unguided Media (Wireless). Switching: Circuit Switched, Datagrams, Virtual Circuit Networks, Structure of a Switch. Telephone Network, Dial-Up MODEMS, Digital Subscriber Line (DSL), Cable TV Networks, Cable TV for Data Transfer.

UNIT-IV

Error Detection and Correction: Introduction, Block Coding, Linear Block Codes, Cyclic Codes, Checksum. Data Link Control: Framing, Flow and Error Control, Protocols, Noiseless Channels, Noisy Channels, HDLC, Point-To-Point Protocol. Multiple Access: Random Access, Controlled Access, Channelization. Wired LANs: IEEE Standards, Standard Ethernet, Changes in the Standard, Fast Ethernet, Gigabit Ethernet: Wireless LANs: IEEE 802.11, Bluetooth.

UNIT-V

Connecting LANs: Connecting Devices, Backbone Networks, Virtual LANs. Wireless LANs: Cellular Telephony, Satellite Networks. SONET: Architecture, SONET Layers, SONET Frames, STS Multiplexing, SONET Networks, Virtual Tributaries. Virtual-Circuit Networks. Frame Relay, ATM, ATM LANs,

TEXT BOOK: Data Communications and Networking, 4/e, Forouzan, B. (TMH)

Semester-IV COMP. SC. (Hons)

CC-X - DESIGN AND ANALYSIS OF ALGORITHMS

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

UNIT-I

Analysis and Design of Algorithm (Case study insertion sort and merge sort) Asymptotic Analysis, Divide and Conquer, Recurrence Relations, Strassen's Matrix Multiplication.

UNIT-II

Sorting: Quick sort, heap sort, Counting sort, lower bound for sorting, Randomized quicksort, Order Statistics.

UNIT-III

Amortized Analysis (Aggregate analysis, Accounting analysis, Potential analysis), 2-3-4 tree Advanced Data structure: Fibonacci heap, Redblack tree, hashing, data structure on disjoint set, Scicinet Data Structure.

UNIT-IV

Dynamic Programming: Matrix Chain multiplication, LCS, TSP, Branch and Bound. Greedy Algorithm: MST: Kruskal, Prim's, Dijkstra Algorithm, Huffman Coding, Maxflow matching, Computational geometry: Convex Hull, 0-1-knapsack, fractional knapsack, Back tracking (4-Queen Prob.)

UNIT-V

Complexity Class: P, PSPACE, NP, NP-Hard, NP Complete, Satisfiability, Clique, Vertex Cover, Independent set, Exact cover, Graph Coloring, Hamiltonian, Cycle Matching. Approximation Algorithm: Vertex Cover, TSP, Independent Set, Sum of subset

Text Book:

Introduction To Algorithm: Corman, Leisenm, Rives & Stein

SEM-III Comp Sc.DSC-1/2/3-III(for Gen) IS SAME AS CC-V

SEM-IV Comp Sc DSC-1/2/3-IV(for Gen) IS SAME AS CC-VI

SEMESTER –IV COMP.SC.
SEC-II Information Technology.

Time-2hrs.

F.M.-40 +10(Int.)

Credit -2

UNIT-1

Data ,Information and Computer Organization:-Introduction, Type of Data, A simple Model of a computer, Data processing Using a Computer

Data storage:-Introduction, Memory Organization, Physical Devices used as Memory Cell, Random access Memory, Read only Memory, Secondary Memory, Compact Disk Read only Memory(CDROM) Flassh Memory.

Central Processing Unit:-Introduction, The Structure of a central Processing Unit, Inteconnection of CPU with Meomory and I/O Units.

UNIT-2

Data Organisation and Software:-

Data Organisation: Intriduction, Organizing a Database, Structure of a Database, Database Management System, Example of Database Design, Non-text Database, Archiving Database.

Processing Numerical and text Data: Introduction, use of Spreadsheets, Numerical Computation Examples, Microsoft Word, Notepad, Learning Power point presentation.

Software:- Introduction Operatig system, Flowcharts, Programming Language Classification of Programming Languages.

Unit -3

Internet technology:

Computer Networks :introduction, local area network(LAN),applications of LAN,wide area networks(WAN),Internet,naming computers connected to internet,the future of internet technology.

Some internet application: Introduction ,email ,information browsing service, the world wide web, Information retrieval from the world wide web, other facilities provided by browser, audio on the internet, pictures, animations, and video via internet

Unit-4

Societal impacts and ethics of information technology:

Introduction, privacy, security and integrity of information, disaster recovery,Intellectual property rights,carriers in information technology and other relevant applications in business,science, judicial system, Healthcare system and engineering.

Text Books:-

1. V. Rajaraman , introduction to Informrmation Technology, Prentice Hall of India,NewDelhi
2. P.K. Sinha and P.Sinha, Computer Fundamentals, BPB Publication.

Reference Books:-

1. V. RajaRaman, fundamentals of computers, Prentice hall of India, NewDelhi
2. M.M Mano CComputer system Architecture, Prentice hall of India, NewDelhi, - 1995.
- 3 Efraime Turban, R.Kelly Rainer, and Richard E.Potter, Introduction to Information Technology ,Wiley.
4. Raj Kamal,Internet and Web Technology, Tata McGrow Hill Education.
5. J.P.Glaser and Cloudia Salzberg, the Strategic Application of Information Technology in Health Care orgabization, Jossey Bass.
6. R. Elmasri and S.B. Navathe, Fundamentals of Data base system, Addison Wisley

SEMESTER –IV COMP.SC. (HONS.)
SEC-II Object Oriented Analysis & Design using UML
Time-2hrs. F.M.-40 +10(Int.) Credit -2

UNIT I: Introduction

Importance of modeling, principles of modeling, Overview of UML, Conceptual model of the UML, Software development life cycle.

UNIT II: Basic Structural Modeling

Building blocks of UML, Classes-Common modeling techniques like the vocabulary of a system, the distribution of responsibilities in a system non software things, relationship common modeling techniques like simple dependencies ,single inheritance, structural relationships, common mechanisms interfaces, types and roles, common modeling techniques like the seams in a system, static and dynamics types, packages-group elements, instance concrete instance and prototypical instances

III: Diagrams and Basic Behavioral Modeling

Terms and Concepts of diagram, Common Modeling techniques- modeling different views of a system, modeling different levels of Abstraction, modeling complex views , Interactions, use cases, use case diagrams, object diagrams, interaction diagrams- modeling flow of control, activity diagrams, process and threads, time and space, state chart diagram, and collaboration diagram

UNIT IV: Architectural Modeling

Components-common modeling techniques like executables and libraries, tables, files, documents, component diagrams sources code, executable release, physical database, adaptable systems, forwards and reverse engineering, deployment processors and devices, distribution of component, deployment diagram- , xecuLable Dataoase', Adapiable Systems, -forwardreverse engineering, deployment-processor, and devices, distribution of components; deployment embedded system, client/server system, fully distributed system, forward and reverse engineering, collaboration.

Text Books:

1. Grady Booch, James Rarnbaugh, Ivar Jacobson, "the unified modeling Language user guide", Person Education, 2nd edition,2004

Reference Books:

2. Michael R Blaha and James R. Rambaugh, "Object oriented modeling and design with UML", Second edition ,2005, Pearson Education ,India NewDelhi

3. Mark Priestley, "Practical object oriented design with UML", 2nd Edition, 2006, McGraw Hill Education, India , NewDelhi

4. Mike O'Docherty,"Object Oriented analysis and design understanding system development with UML 2.0" ,2005, wiley India Pvt. Ltd.,NewDelhi

3rd Semester Comp. Sc. (Hon's)
LAB (CC 3.1): Operating System
F.M-25 (Lab+Viva+Record=15+6+4)

1. List of UNIX/LINUX commands: mkdir, cd, mv, rmdir, pwd, chmod, ls, cp, rm, more, less, cat, wc, cal, head, tail, ping, sleep, comm, cmp, diff, uniq, sort, grep, cut, paste, banner, echo, set
2. Shell Programming
 - (i) Write a program to generate the Fibonacci series.
 - (ii) Write a program to generate prime numbers upto 100.
 - (iii) Write a program to check the string/number as palindrome.
 - (iv) Write a program to find the factorial of given positive number.
 - (v) Write a program to find HCF and LCM of given two numbers.
 - (vi) Write a program to find the sum of digits of given number.
 - (vii) Write a program to find the reverse of given number.
 - (viii) Write a program to check that given year is a leap year or not.
 - (ix) Write a program to find the sum of first n natural numbers.
 - (x) Write a program to find the largest /smallest of three given numbers.

LAB (CC 3.2): Database Management System
F.M-25(Lab+Viva+Record=15+6+4)

1. Familiarity with Oracle commands: SQL command:
DML, DDL, DCL, TCL
2. PL/SQL programming with control structure, procedure, functions, package, Triggers.
 - (i) Write a PL/SQL program to find the largest/smallest number among three numbers.
 - (ii) Write a PL/SQL program to find the sum of numbers from 1 to n.
 - (iii) Write a PL/SQL program to print the factorials of number.
 - (iv) Write a function that accept two numbers A&B, sum/subtract it and store in C.
 - (v) Write a function which accepts two numbers and multiply.
 - (vi) Write a function which will accept a 4-digit year and will print whether the year is leap year or not.
 - (vii) Write a procedure that accepts two numbers add them and print.
 - (viii) Write a procedure which accepts two numbers and prints its sum, difference, multiplication and division.
 - (ix) Write the steps to create the packages.
 - (x) Write a Trigger which will convert ENAME to upper case at time of insertion of record in EMPLOYEE table.

LAB (CC 3.3): Discrete Structure
F.M-25 (Lab+Viva+Record=15+6+4)

1. Show that if any numbers from 1 to 8 are chosen, then two of them will add to 9. (Pigeonhole Principle)
2. How many friends must you have to guarantee at least five of them will have birthdays in the same month?
3. In how many ways can seven people be seated in a circle?
4. Find out minimum spanning trees from a connected weighted graph.
5. Find out the shortest path between two distinct vertices of a weighted connected graph.

4th Semester Comp. Sc. (Hon's)
LAB (CC -VIII): JAVA PROGRAMMING
F.M-25 (Lab+Viva+Record=15+6+4)

1. WAP to find the largest of n natural numbers.
2. WAP to convert the given temperature in Fahrenheit to Celsius and Celsius to Fahrenheit and display the values in tabular form.
3. WAP to find whether a given number is prime or not.
 - a. -WAP to find factorial of given number.
4. WAP to print the sum and product of digits of an integer and reverse the integer.
5. WAP to check the string is palindrome or not.
6. Write a program to create an array of 10 integers. Accept values from the user in that array input another number from user and find out how many numbers are equal to the number passed. How many are greater and how many are less than the number passed.
7. WAP for the following matrix operations.
 - (i) Addition of two matrices
 - (ii) Multiplication of two matrices
 - (iii) Transpose of matrix
8. WAP that compute the area of circle, rectangle and square using function overloading.
9. Develop an Applet that receives two numbers as input from the user and then displays the sum of two in screen.

LAB (CC -IX): COMPUTER NETWORK

F.M-25 (Lab+Viva+Record=15+6+4)

1. Simulate Checksum Algorithm.
2. Simulate CRC Algorithm.
3. Simulate Stop & Wait Protocol.
4. Simulate Go-Back-N Protocol.
5. Simulate Selective Repeat Protocol.
6. Simulate the structure of Cellular Telephony network.
7. Simulate the structure of Satellite Network.

LAB (CC -X): DESIGN AND ANALYSIS OF ALGORITHMS

F.M-25 (Lab+Viva+Record =15+6+4)

1. Implement insertion sort (The program should report the number of comparison)
2. Implement Merge Sort (The program should report the number of comparison)
3. Implement Heap Sort (The program should report the number of comparison)
4. Implement Randomized Quick sort (The program should report the number of comparison)
5. Implement Radix sort
6. Implement searching techniques.
7. Implementation of single, Double and circular linked list.
8. Creation and traverse of BST.
9. Implement Kruskal Algorithm.
10. Implement Prims Algorithm.

B. Sc (Computer Science) Hon's

SEMESTER-V, COMP SC.(H)-CC-11

(TH.): PRINCIPLES OF COMPILER DESIGN

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

Unit –I: Introduction to Compiler, Compiler, Analysis of the source program, phases of compiler, cousins o the compiler, grouping of phases, compiler construction tools, simple one pass compiler –overview, syntax definition, syntax directed translation, parsing, lexical analysis, incorporating symbol table.

Unit-II: Lexical analysis: role of lexical analyzer, input buffering, specification of tokens; recognition of token, finite automata, from a regular expression to NFA (Non-Deterministic Finite Automata), Design of a lexical analyzer generator, optimization of DFA (Deterministic Finite Automata) based pattern matches.

Unit-III: Syntax analysis: role of parser, top-down parsing, bottom-up parsing, operator precedence parsing, LR parsers, using ambiguous grammar, syntax directed translation, syntax directed definitions, construction of syntax trees.

Unit –IV: Runtime environment: source language issues, storage organization, storage allocation strategies, storage allocation in C, parameter passing, intermediate code generation, intermediate languages, declaration , assignment statements.

Unit-V: Code generation: issues in the design of a code generator, runtime storage management, basic blocks and flow graph, register allocation and assignment, DAG representation of basic blocks, generating codes from DAGs, introduction to code generation: introduction, principle sources of optimization, peephole optimization, and optimization of basic blocks.

Text Book: Alfred Aho, Jeffrey D. Ullman, Ravi Sethi, “Compiler Design- Principles Techniques and Tool”, 2nd edition Pearson Education, 2011.

V- Semester Comp. Sc. (Hon's)
LAB(CC-11): Principle of Compiler Design
F.M-25(Lab+Viva+Record=15+6+4)

1. Show the output of a C/C++ compiler, in any typical assembly language, for the following C/C++ input string:
`while (x<a+b) x = 2*x;`
2. Show the compiler output and the interpreter output for the following C/C++ source code:
`for (i=1; i<=4; i++) cout << i*3;`
3. Write C/C++ program to translate n infix expression into postfix form.
4. Construct a regular expression and DFA
 - (i) Which accept the string of even number of a's and b's over {a,b}?
 - (ii) Which accept the string containing 'bab' over {a,b,c}?
 - (iii) Which accept the strings containing an odd number of zeros and an even number of ones?
5. Construct NFA which accepts a string
 - (i) That contains any number of 'a' ending with 'b' over {a,b}
 - (ii) That starting with 'b' followed by two 'a' over {a,b}

B. Sc (Computer Science) Hon's

SEMESTER-V, COMP SC.(H)-CC-12

(TH.): SOFTWARE ENGINEERING

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

Unit-I : Professional Software Development, Software Engineering Ethics, Software Processes, Software Pro-cess Models, Process Activities, Coping with Change, The Rational Unified Process, Agile Software Development, Agile Methods, Plan-Driven and Agile Development, Extreme Programming, Agile Project Management, Scaling Agile Methods.

UNIT-II: Requirements Engineering, Functional and Non-Functional Requirements, The Software Requirements Document, Requirements Specification, Requirements Engineering Processes, Requirements Elicitation and Analysis, Requirements Validation, Requirements Management, System Modelling, Context Models, Interaction Models,

Structural Models, Behavioural Models, Model-Driven, Engineering, Architectural Design, Architectural Design Decisions, Architectural Views, Architectural Patterns, Application Architectures.

UNIT-III: Design and Implementation: Object-Oriented Design using the UML, Design Patterns, Implementation Issues, Open Source Development, Software Testing: Development Testing, Test-Driven Development, Release Testing, User Testing, Software Evolution: Evolution Processes, Program Evolution Dynamics, Software Maintenance, Legacy System Management, Dependability and Security.

UNIT-IV: Socio-technical Systems: Complex Systems, Systems Engineering, System Procurement, System Development, System Operation. Dependability and Security: Dependability Properties, Availability and Reliability, Safety, Security. Dependability and Security Specification: Risk-Driven Requirements, Specification, Safety Specification, Reliability Specification, Security, Specification, Formal Specification.

UNIT-V: Dependability Engineering: Redundancy and Diversity, Dependable Processes, Dependable Systems Architectures, Dependable Programming. Security Engineering: Security Risk Management, Design for Security, System Survivability. Dependability and Security Assurance: Static Analysis, Reliability Testing, Security Testing, Process Assurance, Safety and Dependability Cases.

Recommended Books:

1. Ian Sommerville, Software Engineering, 9/e, Pearson Education.
2. R. Mall, Fundamentals of Software Engineering, 3/e, PHI.
3. R.S. Pressman, Software Engineering, A Practitioners Approach, 7/e, McGraw-Hill, 2009.
4. K.K. Aggarwal and Y. Singh, Software Engineering, 2/e, New Age International Publishers, 2008.

V-Semester Comp. Sc. (Hon's)
LAB(CC-12): Software Engineering
F.M-25 (Lab + Viva + Record=15+6+4)

1. Develop SRS (Software Requirement Specification) for Railway Reservation System (RRS) and

- i. Draw ERD/DFD up to appropriate levels for RRS.
 - ii. Develop a Modular Design for RRS.
1. Develop SRS (Software Requirement Specification) for Library Information System (LIS) and
 - i. Draw ERD/DFD up to appropriate levels for LIS.
 - ii. Develop a Modular Design for LIS.
 2. Develop SRS (Software Requirement Specification) for Hospital Management System (HMS) and
 - i. Draw ERD/DFD up to appropriate levels for HMS.
 - ii. Develop a Modular Design for HMS.

B. Sc (Computer Science) Hon's

SEMESTER-V, COMP SC.(H) DSE-I

(TH.)INFORMATION SECURITY

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40(Th.) + (Pr.)

UNIT-I: Introduction: Security, Attacks, Computer Criminals, Security Services, and Security Mechanisms. Cryptography: Substitution ciphers, Transpositions Cipher, Confusion, diffusion, Symmetric, Asymmetric Encryption. DES Modes of DES, Uses of Encryption, Hash function, key exchange, Digital Signatures, Digital Certificates.

UNIT-II: Program Security: Secure programs, Non malicious Program errors, malicious codes virus, Trapdoors, Salami attacks, Covert channels, Control against program.

UNIT-III: Threats: Protection in OS: Memory and Address Protection, Access control, File Protection, User Authentication. Database Security: Requirements, Reliability, Integrity, Sensitive data, Inference, Multilevel Security.

UNIT-IV: Security in Networks: Threats in Networks, Security Controls, firewalls, Intrusion detection systems, Secure e-mails.

UNIT-V: Administrating Security: Security Planning, Risk Analysis, Organisational Security Policy, Physical Security. Ethical issues in Security: Protecting Programs and data. Information and law.

Recommended Books:

1. C. P. Peeger, S. L. Peeger; Security in Computing, PHI, 2006.

2. W. Stallings; Network Security Essentials: Applications and Standards, 4/E, 2010.

V-Semester Comp. Sc. (Hon's)
LAB (DSE -I): Information Security
F.M-25 (Lab+Viva + Record=15+6+4)

1. Demonstrate the use of network tools: ping, ipconfig, ifconfig, tracert, arp, netstar, who is.
2. Use of password cracking tools: John the ripper, ophcrack, verifies the strength of password using the tools.
3. Perform the encryption and decryption caesar cipher; write a script for performing these operations.
4. Perform the encryption and decryption of a rail fence cipher; write a script for performing these operations.
5. Demonstrate a of digitally signed documents.
6. Demonstrate a sending of a protected worksheet.

B. Sc (Computer Science) Hon's
SEMESTER-V, COMP SC.(H) DSE-II
(TH.) MICROPROCESSOR

Time – 3 hrs. F.M. – 100 [60 (Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40 (Th.) + (Pr.)

UNIT-I: Architecture of 8086: Introduction to computers, Digital computers, microprocessors, 8086 Internal Architecture , memory organisation, input and output structure, programmable H/w registers, addressing modes. Levels of programming.

UNIT-II: 8086 Instructions: Introduction, Assembler instruction format, Data Transfer Instructions, Arithmetic and Logical Instructions, Branch Instructions, Processor control instructions, string operation instructions.

UNIT-III: Assembler directors, introduction, symbols, variables, and constants, data definition and storage allocation directives, Program organisation directives, Assembly Language Programming: Introduction, Program segments, procedures, program structures, Input-Output

structure and programming, program development tools, program development process.

UNIT-IV: An Introduction to Processor Design: Processor architecture and organization , Abstraction in hardware design, MUO - a simple processor, Instruction set design ,Processor design trade-offs ,The Reduced Instruction Set Computer, Design for low power consumption .The ARM Architecture: The Acorn RISC Machine ,Architectural inheritance, The ARM programmer's model, ARM development tools.

UNIT-V: Architectural Support for High-Level Languages: Abstraction in software design, Data types, Floating-point data types, The ARM floating-point architecture, Expressions, Conditional statements, Loops , Functions and procedures, Use of memory, Run-time environment, Examples and exercises.

Recommended Books:

1. Steve Furber: ARM System-On-Chip Architecture.
2. K.R Venugopal, Raj Kumar, “Microprocessor X86 programming”, BPB publications, chapters-1(1.1 -1.9),chapter-2(2.1-2.3) ,chapter-3 (3.1-3.7), chapter-4(4.1-4.4,4.6-4.8)

**V- Semester Comp. Sc.(Hon's)
LAB (DSE -II): Microprocessor
F.M-25 (Lab+Viva + Record=15+6+4)**

1. Write an Assembly language programming for sum of n numbers.
2. Write an Assembly language programming for 32-bit addition of two numbers.
3. Write an Assembly language programming for 32-bit subtraction of two numbers.
4. Write an Assembly language programming to check if the number is odd or even.
5. Write an Assembly language programming to find largest of n numbers.
6. Write an Assembly language programming for product, square and cube of two positive numbers.
7. Write an Assembly language programming to compute the factorial of a number.

8. Write an Assembly language programming for GCD of two positive integer numbers.

V –SEM BSc Computer Science (Pass)

SEC-3: COMPUTER NETWORK

Time – 2 hrs

F.M. – 50 [40 (Sem) +10 (Int.)]

UNIT-I: Introduction: Data Communications, Networks, The Internet, Protocols and Standards. Network Models: Layered Tasks, The OSI Model, Layers in the OSI Model, TCP/ IP Protocol Suite, Addressing.

UNIT-II: Data and Signals: Analog and Digital, Periodic Analog Signals, Digital Signals, Transmission Impairment, Data Rate Limits, Performance. Digital Transmission: Digital-To-Digital Conversion, Analog-To-Digital Conversion, Transmission Modes. Analog Transmission: Digital-To-Analog Conversion, Analog-To-Analog Conversion.

UNIT-III: Multiplexing and Spreading: Multiplexing, Spread Spectrum. Transmission Media: Guided Media, Unguided Media (Wireless). Switching: Circuit Switched, Datagrams, Virtual Circuit Networks, Structure of a Switch. Telephone Network, Dial-Up MODEMS, Digital Subscriber Line (DSL), Cable TV Networks, Cable TV for Data Transfer.

UNIT-IV: Error Detection and Correction: Introduction, Block Coding, Linear Block Codes, Cyclic Codes, Checksum. Data Link Control: Framing, Flow and Error Control, Protocols, Noiseless Channels, Noisy Channels, HDLC, Point-To-Point Protocol.

Recommended Books:

1. B. A. Forouzan, Data Communications and Networking, 4/e, THM ,2007
2. A. S. Tanenbaum, & David J. Wetherall, Computer Networks, 5/e, Pearson

DSE-A/B.C-I same as III Sem Comp. Sc (Hons) CC-VI
GE-I same as I Sem. Comp. Sc. (Hons) GE-I

B. Sc (Computer Science) Hon's

SEMESTER-VI, COMP SC.(H)-CC-13

(TH.): ARTIFICIAL INTELLIGENCE

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40 (Th.) + (Pr.)]

UNIT-I: Intelligent Agents, Solving problems by searching, Uninformed search strategies(BFS, DFS, DLS,IDS, BD and Uniform cost search), Informed search and exploration (Greedy Best first, A* and its variations) Constraint satisfaction Problems, Adversarial search(Alpha-beta pruning).

UNIT-II: Knowledge and reasoning, logical agent (Wumpus world), Propositional logic, First order logic, Inference in first order logic(Forward chaining, backward chaining, Resolution) , Knowledge representation.

UNIT-III: Planning, Partial-Order planning, Planning Graphs, Planning and acting in the real world, Uncertain knowledge and reasoning.

UNIT-IV: Learning from Observations, Decision trees, Neural network (Multilayer), Reinforcement Learning.

UNIT-V: NLP Communication, A formal grammar for a fragment of English, Syntactic analysis (chat parsing), semantic Interpretation, Ambiguity of grammar, Machine Translation.

Recommended Books:

1. Stuart Russell and Peter Norvig, ARTIFICIAL INTELLIGENCE A MODERN APPROACH, 2/e, PHI.
2. D.W. Patterson, Introduction to A.I and Expert Systems, PHI, 2007.
3. Rich & Knight, Artificial Intelligence, 2/e, Tata McGraw Hill, 1991

VI- Semester Comp. Sc.(Hon's)

LAB (CC -13): Artificial Intelligence

F.M-25(Lab+Viva+Record=15+6+4)

1. Write a Lisp/PROLOG programming for finding factorial of given number.
2. Write a Lisp/PROLOG programming to find the area of circle of given radius.
3. Write a Lisp/PROLOG programming to find the sum, product, division, subtraction of two numbers and increment 1st number by 3 and 2nd by 4.

4. Write a Lisp/PROLOG programming for comparison between two numbers i.e. =, <, >, >=, <= and find maximum, minimum between two numbers.
5. Write a Lisp/PROLOG programming to find the day in a week.
6. Write a Lisp/PROLOG programming to print a number between 10 to 18.
7. Write a Lisp/PROLOG programming to find the square of numbers from 1 to 9.
8. Write a Lisp/PROLOG programming for average of any five numbers.

SEMESTER-VI, COMP SC.(H)-CC-14

(TH.): COMPUTER GRAPHICS

Time – 3 hrs. F.M. – 100 [60(Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40 (Th.) + (Pr.)]

UNIT-I: Computer Graphics: A Survey of Computer graphics, Overview of Graphics System: Video Display Devices, Raster-Scan Systems, Input Devices, Hard-Copy Devices, Graphics Software, and Introduction to OpenGL. Graphics Output Primitives: Point and Lines, Algorithms for line, circle & ellipse generation, Filled-Area Primitives. Attributes of Graphics Primitives: Point, line, curve attributes, fill area attributes, and fill methods for areas with irregular boundaries, Anti-aliasing.

UNIT-II: Geometric Transformations (both 2-D & 3-D): Basic Geometric Transformations, Matrix Representation and Homogeneous Coordinates, Composite Transformations, Inverse Transformations, Other Transformations (Reflection, shear), Transformation between coordinate systems, Affine Transformations. Two Dimensional Viewing: Viewing pipeline, Clipping Window, Normalization & Viewport coordinate Transformations, Clipping Algorithms: Point clipping, Line clipping and Polygon clipping. Three Dimensional Viewing: 3-dimensional Viewing Concepts, Viewing pipeline, Projection Transformations (Orthogonal, Oblique parallel, Perspective), Clipping Algorithms.

UNIT-III: Three Dimensional Object Representations: Curved Surfaces, Quadratic Surfaces, Spline Representations, Bezier Spline Curves and Surfaces, B-Spline Curves and Surfaces, Octrees, BSP Trees, Fractal Geometry Methods, Gamma correction.

UNIT-IV: Visible Surface Detection Methods: Classification of Visible-Surface Detection Algorithms, Back-Face Detection, Depth-Buffer

method, A-Buffer Method, Scan line and Depth Sorting, Area subdivision Method, Ray Casting Method.

UNIT-V: Illumination Models: Basic Illumination Models, Displaying light Intensities, Halftone Patterns and Dithering techniques, Polygon-Rendering Methods (Gouraud Shading, Phong Shading), Ray-Tracing Methods (Basic Ray-Tracing Algorithm, Ray-Surface Intersection Calculations). Computer Animation, Hierarchical Modelling (introductory idea only).

Recommended Books:

1. Donald Hearn & M. Pauline Baker, Computer Graphics with OpenGL, Pearson Education.
2. A.V. Dan, F.H. Jones, J.D. Foley, S.K. Feiner, Computer Graphics Principles & Practices in C, 2/e, Pearson.

VI- Semester Comp. Sc. (Hon's)
LAB (CC -14): Computer Graphics
F.M-25 (Lab + Viva + Record = 15+6+4)

1. Write a Program in C/C++ to implement (Draw) a Bresenham's line generation algorithm.
2. Write a Program in C/C++ to implement (Draw) a DDA Line- drawing algorithm.
3. Write a Program in C/C++ to implement (draw) a circle by midpoint circle algorithm.
4. Write a Program in C/C++ to implement (Draw) a Bresenham's Circle generation algorithm.
5. Write a Program in C/C++ to implement (Draw) by midpoint ellipse algorithm.
6. Write a program in C/C++ to implement (Draw) translation of line.
7. Write a program in C/C++ to implement (Draw) shearing of line.
8. Write a program in C/C++ to implement (Draw) a line clipping by Cohen Sutherland algorithm.

B. Sc (Computer Science) Hon's

SEMESTER-VI, COMP SC.(H) DSE-3

(TH): CLOUD COMPUTING

Time – 3 hrs. F.M. – 100 [60 (Sem) +15 (Int.) + 25 (Pr.)] Credits: -04(Th.) +02(Pr.)

Lectures – 60 [40 (Th.) + (Pr.)

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

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

UNIT-III: Case Studies: Case Study of Service, Model using Google App Engine, Microsoft Azure, Amazon EC2, Eucalyptus.

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

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

Recommended Books:

1. Barrie Sosinsky, Cloud Computing Bible, Wiley-India, 2010.
2. Rajkumar Buyya, James Broberg, Andrzej, M. Goscinski, Cloud Computing Principles & Paradigms, Wiley-2011.
3. Nikos Antonopoulos, Lee Gillam, Cloud Computing: Principles, Systems and Applications, Springer, 2012.

VI- Semester Comp. Sc. (Hon's)
LAB (DSC -III): CLOUD COMPUTING
F.M-25 (Lab +Viva + Record=15+6+4)

1. Create virtual machine that access different program on same platform.
2. Create virtual machine that access different program on different platforms.
3. Working on tools used in cloud computing online:
 - i) Storage
 - ii) Sharing of data
 - iii) Manage your calendar to list
 - iv) A document editing tool

4. Exploring Google cloud.
5. Exploring Microsoft Cloud.
6. Exploring Amazon Cloud.

B. Sc (Computer Science) Hon's

SEMESTER-V, COMP SC. (H) DSE-4

Project work (Compulsory)

Time – 3 hrs.

F.M. – 100 [60 (Dissertation) +25 (Presentation.) + 15(Viva)]

A student has carry on project work in the last semester of three year degree honours course choosing any topic from

1. Scientific Computing
2. Developing application software using Java, DBMS, C/C++ and Data structure.
3. Networking and Webpage Designing using Java, Internet Technology.

The student has to submit dissertations and present a seminar on his/her work (submit by hardcopy of project work)

Instruction:

1. Project topic should be decided by HOD and faculty of department.
2. Internal and External examiner should appointed by the University to evaluate the project work.

V –SEM BSc Computer Science (Pass) SEC-4: SOFTWARE ENGINEERING

Time – 2 hrs

F.M. – 50 [40 (Sem) +10 (Int.)]

Unit-I : Professional Software Development, Software Engineering Ethics, Software Processes, Software Pro-cess Models, Process Activities, Coping with Change, The Rational Unified Process, Agile Software Development, Agile Methods, Plan-Driven and Agile Development, Extreme Programming, Agile Project Management, Scaling Agile Methods.

UNIT-II: Requirements Engineering, Functional and Non-Functional Requirements, The Software Requirements Document, Requirements Specification, Requirements Engineering Processes, Requirements Elicitation and Analysis, Requirements Validation, Requirements Management,

UNIT-III: System Modelling, Context Models, Interaction Models, Structural Models, Behavioural Models, Model-Driven, Engineering, Architectural Design, Architectural Design Decisions, Architectural Views, Architectural Patterns, Application Architectures. Design and Implementation: Object-Oriented Design using the UML, Design Patterns, Implementation Issues, Open Source Development,

UNIT-IV: Software Testing: Development Testing, Test-Driven Development, Release Testing, User Testing, Software Evolution: Evolution Processes, Program Evolution Dynamics, Software Maintenance, Legacy System Management, Dependability and Security.

Recommended Books:

1. Ian Sommerville, Software Engineering, 9/e, Pearson Education.
2. R. Mall, Fundamentals of Software Engineering, 3/e, PHI.
3. R.S. Pressman, Software Engineering, A Practitioners Approach, 7/e, McGraw-Hill, 2009.
4. K.K. Aggarwal and Y. Singh, Software Engineering, 2/e, New Age International Publishers, 2008.

DSE-A/B/C-II same as CC-III of Comp. SC Hon's SEM –II
GE-II same as GE-II of Comp. Sc. Hons Sem-II