Raja N. L. Khan Women's College (Autonomous)

Syllabus For Four Years B.Sc. Computer Science (Major)

under

National Education Policy (NEP) Framework

w.e.f Academic Year (2024)

Level	SEM	Course Type	Course Code	Course Title	Credit	L-T-P	Marks		
							CA	ESE	TOTAL
B.Sc. (Hons.)	**I**	**SEMESTER-I**							
		Major-1	COSMJ101	T: Computer fundamentals & C programming	4	3-0-1	15	60	75
				P: Programming in C Lab					
		SEC	COSSEC01	Web Designing	3	0-0-3	10	40	50
				Web Designing Lab					
		AEC	AEC01	Communicative English -1 (Common for all Courses)	2	2-0-0	10	40	50
		MDC	MDC01	Multidisciplinary Course -1 (Environmental Studies)	3	3-0-0	10	40	50
		VAC -1	VAC01	Value Added Course-01 Physical Fitness and Community Service	2	2-0-2	10	40	50
		VAC -2	VAC02	Value Added Course-02 Mental Health and Well Being	2	2-0-2	10	40	50
		Minor-I	COSMI01	Mathematics	4	3-0-1	15	60	75
		Semester-I Total			20				400

SEM	Course Type	Course Code	Course Title	Credit	L-T-P	Marks		
						CA	ESE	TOTAL
II	**SEMESTER-II**							
	Major-1I	COSMJ201	T: Data Structures P: Data Structures Lab	4	3-0-1	15	60	75
	SEC 02	COSSEC02	T: Computing Paradigm with Python Programming P: Python Programming Lab	3	0-0-3	10	40	50
	AEC	AEC02	MIL (Modern Indian Language)	2	2-0-0	10	40	50
	MDC	MDC02	Disaster management	3	3-0-0	10	40	50
	VAC -1	VAC03	Indian constitution and ethics	2	2-0-2	10	40	50
	VAC -2	VAC04	Understanding India	2	2-0-2	10	40	50
	Minor-II		Mathematics	4	3-0-1	15	60	75
	CS		Community Service	2		10	40	50
	Semester-II Total			20				450

Level	SEM	Course Type	Course Code	Course Title	Credit	L-T-P	Marks		
							CA	ESE	TOTAL
B.Sc. (Hons.)	III	**SEMESTER-III**							
		Major-III	COSMJ301	Digital Electronics	4	3-0-1	15	60	75
				Digital Electronics Lab					
		Major-IV	COSMJ302	Object Oriented Programming with Java	4	3-0-1	15	60	75
				Programming in Java Lab					
		SEC -3	COSSEC03	Programming in MATLAB	3	0-0-3	10	40	50
				Programming in MATLAB Lab					
		AEC	AEC03	Communicative English - 1 (Common for all Courses)	2	2-0-0	10	40	50
		MDC	MDC03	Digital Fluency & Artificial Intelligence	3	3-0-0	10	40	50
		Minor-III	**COSMI01**	Physics/Biotechnology/ Economics	4	3-0-1	15	60	75
		Semester-III Total			20				475

Level	SEM	Course Type	Course Code	Course Title	Credit	L-T-P	Marks		
							CA	ESE	TOTAL
B.Sc. (Hons.)	IV	**SEMESTER-IV**							
		Major-V	COSMJ401	Computer organization and Architecture	4	3-0-1	15	60	75
				Computer Architecture Lab					
		Major-VI	COSMJ402	Operating Systems	4	3-0-1	15	60	75
				Operating Systems Lab					
		Major-VII	COSMJ403	Data Communications & Computer Networks	4	3-0-1	15	60	75
				Data Communications & Computer Networks Lab					
			MIL	**MIL**	2		10	40	50
		CS		Community Service	2		10	40	50
		Minor-III	**COSMI02**	Physics/ Biotechnology	4	3-0-1	15	60	75
		Semester-IV Total			20				400

Level	SEM	Course Type	Course Code	Course Title	Credit	L-T-P	Marks		
							CA	ESE	TOTAL
B.Sc. (Hons.)	V	**SEMESTER-V**							
		Major-VIII	COSMJ501	Design and Analysis of Algorithms Design and Analysis of Algorithms Lab	4	3-0-1	15	60	75
		Major-IX	COSMJ502	Software Engineering Software Engineering Lab	4	3-0-1	15	60	75
		Major-X	COSMJ503	Database Management System Database Management System Lab	4	3-0-1	15	60	75
		Major-XI	COSMJ504	Discrete Structures	4	3-0-1	15	60	75
		Minor-III	COSMI03	Physics/ Economics/Biotechnology	4	3-0-1	15	60	75
		Semester-V Total			20				375

Level	SEM	Course Type	Course Code	Course Title	Credit	L-T-P	Marks		
							CA	ESE	TOTAL

B.Sc. (Hons.)	**VI**	**SEMESTER-VI**							
		Major-XII	COSMJ601	Machine Learning Machine Learning Lab	4	3-0-1	15	60	75
		Major-XIII	COSMJ602	Theory of Computation	4	4-0-0	10	60	75
		Major-XIV	COSMJ603	Computer Graphics Computer Graphics Lab	4	3-0-1	15	60	75
		Major-XV	COSMJ604	Project- Minor	4	0-0-4	5	70	75
				Summer Internship	2				50
		Minor-IV	COSMI04	Physics /Biotechnology	4	3-0-1	15	60	75
		Semester-VI Total			22				425

COSMJ101: Computer Fundamentals and C programming
(40 Lectures)

LIST OF COURSE OUTCOMES

CO1 Students will be able to develop simple applications in C using basic constructs
CO2 Students will be able to design and implement applications in C using Arrays and Strings
CO3 Students will be able to design and implement applications in C using Functions and Pointers
CO4 Students will be able to develop applications in C using Structures
CO5 Students will be able to design applications using sequential and random access file processing.

Unit-I

Module I: Introduction to Computer systems (3 Lectures)

Generations of Computer (I-V), number system(Binary, Decimal, Hexadecimal ,Octal Numbers, Signed Numbers), Block Diagram of a Computer, Functions of the Different Units Input unit ,Output unit, Memory unit, CPU (ALU+CU).

Module II: Input & Output Devices (2 Lectures)

Input Devices: Keyboard, Point and draw devices, mouse, joystick, track ball, light pen , Data Scanning devices image scanner, OCR, OMR, MICR, Bar code reader, card reader, Voice Recognition Device, Digitizers Output Devices: Monitor, Printer laser printer, dot matrix printer, ink jet printer , Projector

Module III:.Memories (Memory hierarchy) (3 Lectures)

Registers (Types of Registers),Cache Memory, Primary Memory , RAM : How data is stored in a RAM, DRAM and SRAM, ROM: ROM BIOS/ Firmware, Types of ROM Secondary Memories: Hard disk : Structure of a hard disk, how data is stored in a hard disk, concept of tracks, sectors, clusters, cylinders, formatting of hard disk (low level formatting and high level formatting) Floppy (data storage mechanism) CD (data storage mechanism).

Module IV: System Software (4 Lectures)

Operating System , Functions of OS, Types of OS, Program Language Translators : Assembler, Compiler, Interpreter. Utility Programs, Communication Software, Performance Monitoring Software, Application Software. Software hierarchy and dependence between the different layers. Computer Languages: Machine language, Assembly language, High level language.

Unit –II: Programming In C

Module I:

Introduction to Programming (3 Lectures)

Problem analysis, Flowchart, algorithms, Pseudo codes, structured programming, Example of Flowchart and Algorithm representation, Brief History of Development of C language, Features of C language, Process of compiling and running a C program, Overview of Procedural Programming, Performance Analysis and Measurements: Time and Space complexity , Use of main() function.

Module II: Data Types, Variables, Constants, Operators and Basic I/O (4 Lectures)

Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords,Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putchar etc), Formatted and Console I/O (printf(), scanf()),using basic header files (stdio.h, conio.h etc).

Module III: Expressions, Conditional Statements and Iterative Statements (8 Lectures)

Simple Expressions in C (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators, Precedence in Expressions, Conditional Statements (if construct, switchcase 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)

Module IV: Functions and Arrays (9 Lectures)

Utility of functions, Call by Value, Call by address, Functions returning value, Void functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Recursion. Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two- dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays. Different storage classes and scope of variables.

Module V: Derived Data Types (Structures and Unions) (4 Lectures)

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 datamembers as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members, enum.

Module VI: String (1 Lectures)

Character Array, Library functions related to string (strcat(), strcmp(), strcpy(), strlen())

Module VI:Pointers (4 Lectures)

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. Pointer arithmetic. Command Line Arguments.

Module VI: Dynamic Memory Allocation (4 Lectures)

Differentiating between static and dynamic memory allocation, use of malloc, calloc, realloc and free functions, storage of variables in static and dynamic memory allocation

Module VI:File I/O, Preprocessor Directives (4 Lectures)

Opening and closing a file (use of fopen, fclose), Reading and writing Text Files, using fscanf(),fprintf(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros.

Recommended Books

1. Dey, Bandyopadhyay, C Programming Essentials, Pearson Education.
2. Balagurusamy, Programming in ANSI C, Tata McGraw Hill.
3. Kerninghan, Ritchie, The C Programming Language, Pearson.
4. Kanetkar, Let Us C, BPB Publications

C Programming Lab **20 Lectures**

LIST OF COURSE OUTCOMES

CO1 Students will be able to develop C programs for simple applications making use of basic constructs.
CO2 Students will be able to develop C programs for simple applications using Arrays and Strings
CO3 Students will be able to develop C programs involving Functions, Recursion, and Pointers.
CO4 Students will be able to develop C programs involving Structures
CO5 Students will be able to design applications using sequential and random access file processing.

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, breath 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, breath 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.

Recommended Books

1. Dey, Bandyopadhyay, C Programming Essentials, Pearson Education.
2. Balagurusamy, Programming in ANSI C, Tata McGraw Hill.
3. Kerninghan, Ritchie, The C Programming Language, Pearson.
4. Kanetkar, Let Us C, BPB Publications.

COSMJ201:Data Structures

Theory: 40 Lectures

LIST OF COURSE OUTCOMES

CO1 Students will be able to Implement abstract data types for linear data structures.
CO2 Students will be able to Analyze various Stack and queue operations.
CO3 Students will be able to Execute different linear data structures methods trees to problem solutions.
CO4 Students will be able to Apply the different non-linear data structures Graphs to problem solutions.
CO5 Students will be able to Critically analyze the various sorting and hashing algorithms.

1.Arrays (5 Lectures)

Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation)

2.Stacks (5 Lectures)

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

3.Linked Lists (5 Lectures)

Singly, Doubly and Circular Lists (Array and Linked representation);Normal and Circular representation of Stack in Lists; Self Organizing Lists; Skip Lists

4.Queues (5 Lectures)

Array and Linked representation of Queue, De-queue, Priority Queues

5.Recursion (3 lectures)

Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)

6.Trees (10 Lectures)

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).

7.Searching and Sorting (5 Lectures)

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

8.Hashing (5 Lectures)

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.

Reference Books:

1.Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.

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

Data Structures Lab: **20 Lectures**

LIST OF COURSE OUTCOMES:

CO1 Students will be able to Enumerate functions to implement linear and non-linear data structure operations,Perform practical applications of data structures
CO2 Students will be able to Design and develop appropriate linear / non-linear data structure operations for solving a given problem
CO3 Students will be able to Design new solutions for programming problems or improve existing code using learned algorithms and data structures
CO4 Students will be able to Apply the linear / non-linear data structure operations for a given problem based on the user needs
CO5 Students will be able to Use appropriate hash functions that result in a collision free scenario for data storage and retrieval

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 fibonacci 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.

COSMJ301:Digital Electonics:

Theory: 40 Lectures

LIST OF COURSE OUTCOMES

The course should enable the students to:
1. StudyvariousnumbersystemsandtosimplifythemathematicalexpressionsusingBoolean functions – simple problems.
2. Study implementation of combinational circuits.
3. Study the design of various synchronous and asynchronous circuits.
4. LearnabouttheStatereductiontechniques&varioushazardspresentinthecircuit
5. Expose the students to various memory devices and to Design the Digital circuits using HDL programming.

1. Basic Electronics

Introduction to Electronics: Signals, frequency Spectrum of Signals, Analog and Digital Signals, time domain analysis of signal, elementary signals (impulse, step, and ramp), Discrete signals, Amplifiers, Digital logic inverter. Diodes: p-n junction theory, Bipolar junction Transistor (BJTs),Field Effect Transistors (FETs),The ideal Op Amp, different stages of OP Amp.

2. NUMBERSYSTEMS & BOOLEAN ALGEBRA (9 Lectures)

Binary Numbers, Signed-binary numbers, Decimal-to-Binary & Binary-to-Decimal Conversion, Binary Addition, Subtraction, Multiplication and Division, Hexadecimal Number Systems, Logic Gates, Boolean Algebra, De Morgan's Theorems, Laws of Boolean Algebra, Minimization of Boolean expressions – Boolean expressions and Logic Diagrams -Universal building blocks - Negative logic. Logic Simplifications: Truth tables and maps - Sum of products (SOP) and Product of Sum (POS) - Simplification of logic functions using Karanaugh map - Minimization and Quine- McCluskey method of minimization.

3. **COMBINATIONAL CIRCUITS (9 Lectures)**

Arithmetic Circuits: Half Adder, Full Adder, Half Subtractor & Full Subtractor, Number complements. Multiplexer & Demultiplexer, Decoder and Encoder ,Code converters: BCD to Excess3, Gray, Seven Segment Display Conversions – Parity Generator and Checkers.

4**.SYNCHRONOUSSEQUENTIALCIRCUITS (9 Lectures)**

Basic latch circuits - Flip-flops, Truth table and excitation table- Analysis of Clocked Sequential

circuits- Shift Registers. Counters: Synchronous counter design using JK, T, D flip flops, Up-down counter, BCD counter and Ring counters.

5.ASYNCHRONOUSSEQUENTIALLOGIC (9 Lectures)
AnalysisandDesignofAsynchronousSequentialCircuits-ReductionofStateandFlowTables-Multiple Inputs- Race free State Assignment-Hazards

6. LOGIC FAMILIES AND PROGRAMMABLE LOGIC DEVICES (9 Lectures)
HDL Programming: Introduction to HDL Programming, HDL for Combinational Circuits, HDL for sequential logic circuits. Programmable Logic Devices: Programmable Logic Array (PLA) - Programmable Array Logic (PAL) - PROM.

Recommended Books
1. Microelectronics Circuits, A.S Sedra, K.C. Smith, Oxford University Press.
2. Electronics Fundamentals and Applications, D Chattopadhyay and P.C. Rakshit, NewAge International Publications.
3. Mano, Digital Logic and Computer Design, Mano, Pearson.

COSMJ302: Object Oriented Programming with Java
Theory: 40 Lectures

LIST OF COURSE OUTCOMES

CO1 Students will be able to Interpret Java programs using Object Oriented Programming principles
CO2 Students will be able to Explain Java programs with the concepts inheritance and interfaces
CO3 Students will be able to Relate Java applications with threads and generics classes
CO4 Students will be able to Develop Java applications with threads and generics classes
CO5 Students will be able to Develop interactive Java programs using swings, Demonstrate simple Graphical user interface

1. Object-Oriented Programming Overview (4 Lectures)
Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, , encapsulation, inheritance, polymorphism, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

Introduction to Java **(4 Lectures)**

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),

2. Arrays, Strings and I/O **(8 Lectures)**

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.

3. Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata

(14 lectures)

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.

4. Exception Handling, Threading, Networking and Database Connectivity

(15 Lectures)

5. 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.

6. Applets and Event Handling **(15 Lectures)**

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

Reference Books

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

Programming in Java Lab

Practical: 20 Lectures

LIST OF COURSE OUTCOMES

CO1 Students will be able to Develop and implement Java programs for simple applications that make use of classes and to implement Java programs with array list
CO2 Students will be able to Design applications using file processing
CO3 Students will be able to Build software development skills using java programming for real-world applications and
CO4 Students will be able to Apply the concepts of classes, packages, interfaces, exception handling
CO5 Students will be able to Develop applications using generic programming and event handling

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of .lenth in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and String

Bufferclasss like setCharAt(, setLength(), append(), insert(), concat() and equals().

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

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

COSMJ401 : Computer Organization and Architecture **Theory: 40 Lectures**

LIST OF COURSE OUTCOMES

CO1 Students will be able toUnderstand the basics structure of computers, operations and instructions
CO2 Students will be able toDesign arithmetic and logic unit.
CO3 Students will be able toUnderstand pipelined execution and design control unit.
CO4 Students will be able toUnderstand parallel processing architectures.
CO5 Students will be able to Understand the various memory systems and I/O communication.

Introduction

Principles of Computer design - Software, hardware interaction layers in computer architecture basic functional units of a computer, Von-Neumann model of computation, A note on Moore's law, Notion of IPC, and performance. Data representation and basic operations. fixed and floating point representations, character representation. Computer arithmetic – integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift- and add, Booth multiplier, carry save multiplier, etc

Instruction Set Architecture

CPU registers, Control unit, Data path and control path design, Microprogramming V s hardwired control instruction format and encoding, addressing modes ,instruction set, instruction types, instruction decoding and execution, basic instruction cycle, Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer (CISC), RISC-V instructions; X86 Instruction set.

The Processor

Revisiting clocking methodology, Amdahl's law, Building a data path and control, single cycle processor, multi-cycle processor, instruction pipelining, Notion of ILP, data and control hazards and their mitigations.

Memory hierarchy

SRAM/DRAM, locality of reference, Caching: different indexing mechanisms, Trade offs related to block size,associativity, and cache size, Processor-cache interactions for a read/write request, basic optimizations like write- through/write-back caches, Average memory access time, Cache replacement policies (LRU), Memory interleaving.

Storage and I/O Introduction to magnetic disks (notion of tracks, sectors), flash memory. I/O mapped, and memory mapped I/O. I/O data transfer techniques: programmed I/O, Interrupt-driven I/O, and DMA. Superscalar processors and multicore systems Limits of ILP, SMT processors, Introduction to multicore systems and cache coherence issues

Recommended Books:

•M. Mano, Computer System Architecture, Pearson Education 1992

•A. J. Dos Reis, Assembly Language and Computer Architecture using C++ and JAVA, Course Technology, 2004

•W. Stallings, Computer Organization and Architecture Designing for Performance, 8th Edition, Prentice Hall of India,2009

•Carl Hamacher, Computer Organization, Fifth edition, McGrawHill, 2012.

Computer Architecture Lab

Practical: 20 Lectures

LIST OF COURSE OUTCOMES

CO1 Understanding the behavior of Logic Gates, Adders, Decoders, Multiplexers and Flip-Flops.
CO2 Understanding the behavior of ALU, RAM, STACK and PROCESSOR from working modules and the modules designed by the student as part of the experiment.

1. Introduction to Verilog HDL/VHDL
2. Verify the behavior of logic gates using truth tables (AND, OR, NOT, XOR,NAND, NOR)
3. Implementing HALF ADDER, FULL ADDER using basic logic gates
4. Implementing Binary -to -Gray, Gray -to -Binary code conversions
5. Implementing 3-8 line DECODER.
6. Implementing 4x1 and 8x1 MULTIPLEXERS.
7. Verify the excitation tables of various FLIP-FLOPS
8. Design of an 8-bit Input/Output system with four 8-bit Internal Registers.
9. Design of an 8-bit ARITHMETIC LOGIC UNIT.

Design of 24x8 (16 byte) RAM.. Design of 24x8 (16 byte) STACK. . Implementation of a 4-bit PROCESSOR.

References:

1)A Verilog HDL Primer by J. Bhasker Bk&Hardcover; Published by Star Galaxy Press. ISBN: 0-9656277-4-8

2)Verilog HDL : A Guide to Digital Design and Synthesis by Samir Palnitkar Published by Prentice Hall Publication date: March 1996

COSMJ402: Operating Systems **Theory: 40 Lectures**

LIST OF COURSE OUTCOMES

CO1 Students will be able to Gain knowledge about basic concepts and functions of operating system
CO2 Students will be able to Analyse various scheduling algorithms and understand deadlock prevention and
avoidance algorithm
CO3 Students will be able to Compare and contrast various memory management schemes
CO4 Students will be able to Understand the functionality of file systems
CO5 Students will be able to Perform administrative tasks on linux servers and compare ios and android operating systems

1.Introduction (10 Lectures)

Basic OS functions, resource abstraction, types of operating systems–multiprogramming systems, batch systems , time sharing systems; operating systems for personal computers & workstations, process control & real time systems.

2.Operating System Organization (6 Lectures)

Processor and user modes, kernels, system calls and system programs.

3.Process Management (20Lectures)

System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms; concurrent and processes, critical section, semaphores, methods for inter- process communication; deadlocks.

4. Memory Management (10 Lectures)

Physical and virtual address space; memory allocation strategies –fixed and variable partitions, paging, segmentation, virtual memory.

5.File and I/O Management (10 Lectures)

Directory structure, file operations, file allocation methods, device management.

6.Protection and Security (4 Lectures)

Policy mechanism, Authentication, Internal access Authorization.

7. Case study of linux

Recommended Books:

1.A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts,
John Wiley Publications 2008.8th Edition,
2.A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3.G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
4.W. Stallings, Operating Systems, Internals & Design Principles , 5 th Edition, Prentice Hall of India. 2008.
5.M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

Operating Systems Lab

Practical: 20 Lectures

LIST OF COURSE OUTCOMES

CO1 Students will be able to Compare the performance of various CPU Scheduling Algorithms
CO2 Students will be able to Implement Deadlock avoidance and Detection Algorithms
CO3 Students will be able to Implement Semaphore, Create processes and implement IPC
CO4 Students will be able to Analyze the performance of the various Page Replacement Algorithms
CO5 Students will be able to Implement File Organization and File Allocation Strategies

1. Basics of UNIX commands and Implementation of Shell Programming.
2. .Implementation of Process and thread (Life cycle of process): (i) Process creation and Termination; (ii) Thread creation and Termination
3. Implementation of CPU Scheduling. (i) FCFS, (ii) SJF, (iii) Shortest Remaining Time First and (iv) Priority based
4. WRITE A PROGRAM (using fork() and/or exec() commands) where parent and child execute:
    a. a)same program, same code.
    b. b)same program, different code.
    c. c)before terminating, the parent waits for the child to finish its task.
5. WRITE A PROGRAM to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information)
6. WRITE A PROGRAM to report behaviour of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
7. WRITE A PROGRAM to print file details including owner access permissions, file access time, where file name is given as argument.
8. WRITE A PROGRAM to copy files using system calls.
9. Write program to implement FCFS scheduling algorithm.
10. Write program to implement Round Robin scheduling algorithm.
11. Write program to implement SJF scheduling algorithm.
12. Write program to implement non-preemptive priority based scheduling algorithm.
13. Write program to implement preemptive priority based scheduling algorithm.
14. Write program to implement SRJF scheduling algorithm.
15. Write program to calculate sum of n numbers using thread library.
16. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

COSMJ403: Computer Networks Theory:

40 Lectures

LIST OF COURSE OUTCOMES

CO1 Students will be able to Identify various layers of network and discuss the functions of physical layer
CO2 Students will be able to Discuss how data flows from one node to another node with regard to data link
layer
CO3 Students will be able to explain the different services of network layer
CO4 Students will be able to Compare the different transport layer protocols and their applicability based on user requirements
CO5 Students will be able to Describe the working of various application layer protocols and evaluate the performance of network and analyze routing algorithms

1. Introduction to Computer Network (8 Lectures)

Network definition; network topologies; network classifications; network protocol; layered network architecture; overview of OSI reference model; overview of TCP/IP protocol suite.

2. Data Communication Fundamentals and Techniques (10 Lectures)

Analog and digital signal; data-rate limits; digital to digital line encoding schemes; pulse code modulation; parallel and serial transmission; digital to analog modulation-; multiplexing techniques- FDM, TDM; transmission media.

3.Networks Switching Techniques and Access mechanism (10 Lectures)

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

4. Data Link Layer Functions and Protocol (10 Lectures)

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.

5. Multiple Access Protocol and Networks (5 Lectures)

CSMA/CD protocols; Ethernet LANS; connecting LAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways;

6.Networks Layer Functions and Protocols (6 Lectures)

Routing; routing algorithms; network layer protocol of Internet- IP protocol, Internet control protocols.

6. Transport Layer Functions and Protocols (6 Lectures)

Transport services- error and flow control,Connection establishment and release three way handshake;

7. Overview of Application layer protocol (5 Lectures)

Overview of DNS protocol; overview of WWW &HTTP protocol.

Reference Books

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM ,2007.

2. A. S. Tanenbaum: Computer Networks**,** Fourth edition**,** PHI , 2002

Computer Networks Lab: **20 Lectures**

LIST OF COURSE OUTCOMES

CO1 Students will be able to Understand and evaluate the basic layers and its functions in computer networks
CO2 Students will be able to Understand the basics of how data flows from one node to another.
CO3 Students will be able to Analyze and design routing algorithms
CO4 Students will be able to Design protocols for various functions in the network
CO5 Students will be able to Understand the working of various application layer protocols

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 distance vector routing algorithm
4.Simulate and implement Dijkstra algorithm for shortest path routing.
5.Write a program for Creation of sockets,
6.Write a program to communicate between TCP client & server
7.,Write a program to communicate between UDP client & socket
8.Write a socket Program for Echo/Ping/Talk commands

COSMJ501: Design and Analysis of Algorithms

LIST OF COURSE OUTCOMES

CO1 Students will be able to Design algorithms for various computing problems.
CO2 Students will be able to Compare various Brute Force and Divide and Conquer method algorithms
CO3 Students will be able to Analyze the time and space complexity of algorithms.
CO4 Students will be able to Critically analyze the different algorithm design techniques for a given problem
CO5 Students will be able to Modify existing algorithms to improve efficiency

1.Introduction (5 Lectures)
Basic Design and Analysis techniques of Algorithms, Correctness of Algorithm
.**2.Algorithm Design Techniques (8 Lectures)**
Iterative techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.
3.Sorting and Searching Techniques (20 Lectures)

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;

4.Lower Bounding Techniques (5 Lectures)

Decision Trees

5.Balanced Trees

Red-Black Trees

6.Advanced Analysis Technique

Amortized analysis

7.**Graphs**

Graph Algorithms–Breadth First Search, Depth First Search and its Applications, Minimum Spanning Trees.

8. String Processing

String Matching, KMP Technique.

Recommended Books:

1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Introduction to Algorithms, PHI, 3rd Edition 2009

2. Sarabasse& A.V. Gelder Computer Algorithm – Introduction to Design and Analysis, Publisher – Pearson 3rd Edition 1999

Design and Analysis of Algorithms Lab: **20 Lectures**

LIST OF COURSE OUTCOMES

CO1: Develop programs for sorting a given set of elements and analyse its time
complexity.
CO2: Solve and analyse the problems using greedy methods.
CO3: Solve and analyse the problems using dynamic programming.
CO4: Apply backtracking method to solve various problems.
CO5: Apply branch and bound method to solve 0/1 knapsack problem.

1. i. 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.

COSMJ502: Software Engineering Theory:
40 Lectures

LIST OF COURSE OUTCOMES

CO1 Students will be able to Identify the key activities in managing a software project.
CO2 Students will be able to Compare different process models
CO3 Students will be able to Concepts of requirements engineering and Analysis Modeling.
CO4 Students will be able to Apply systematic procedure for software design and deployment.
CO5 Students will be able to Compare and contrast the various testing and maintenance and Manage project schedule, estimate project cost and effort required.

1.Introduction (8 Lectures)
The Evolving Role of Software, Software Characteristics, Changing Nature of Software, Software Engineering as a Layered Technology, Software Process Framework, Framework and Umbrella.

2. Requirement Analysis
Software Requirement Analysis,Initiating Requirement Engineering Process, Requirement Analysis and Modeling Techniques, Flow Oriented Modeling, Need for SRS, Characteristics and Components of SRS .

3.Software Project Management.
Estimation in Project Planning Process, Project Scheduling.

4.Risk Management
software risk, risk identification , risk projection and Risk refinement, RMMM plan

5.Quality Management (8 Lectures)
Quality Concepts, Software Quality Assurance, Software Reviews, Metrics for Process and projects.

6.Design Engineering (10 Lectures)
Design Concepts, Architectural Design Elements, Software Architecture, Data Design at the Architectural Level and Component Level, Mapping of Data Flow into Software Architecture, Modeling Component Level Design.

7.Testing Strategies & Tactics (8 Lectures)
Software Testing Fundamentals, Strategic Approach to Software Testing, Test Strategies for Conventional Software, Validation Testing, System testing,Black-Box Testing, White-Box Testing and their type, Basis Path Testing.

Recommended Books:

1 .R.S. Pressman, Software Engineering: A Practitioner‘s Approach (7th Edition), Mc-Graw-Hill, 2009.
2 .P. Jalote, An Integrated Approach to Software Engineering (2ndEdition), Narosa Publishing House, 2003.
K.K. Aggarwal and Y. Singh, Software Engineering (2nd Edition), New Age International Publishers, 2008.

1.I. Sommerville, Software Engineering (8th edition), Addison Wesley, 2006.
2.D. Bell, Software Engineering for Students (4th Edition), Addison-Wesley, 2005.
3.R. Mall, Fundamentals of Software Engineering (2nd Edition), Prentice-Hall of India, 2004.

Software Engineering Lab: **40 Lectures**

LIST OF COURSE OUTCOMES

CO1 To understand the software engineering methodologies involved in the phases for project development.
CO2 To gain knowledge about open source tools used for implementing software engineering methods.
CO3 To exercise developing product-startupsimplementing software engineering methods.
CO4 Open source Tools: StarUML / UMLGraph / Topcased

To exercise developing product-startups implementing software engineering methods Prepare the following documents and develop the software project startup, prototype model, using software engineering methodology(Open source Tools: StarUML / UMLGraph / Topcased)

- Problem Analysis and Project Planning -Thorough study of the problem – Identify Project scope, Objectives and Infrastructure.
- Software Requirement Analysis – Describe the individual Phases/modules of the project and Identify deliverables. Identify functional and non-functional requirements.

- Data Modeling – Use work products – data dictionary.

- Software Designing – Develop use case diagrams and activity diagrams, build and test class diagrams, sequence diagrams and add interface to class diagrams.

- Prototype model – Develop the prototype of the product.

The SRS and prototype model should be submitted

S. No	Practical Title
1	Problem Statement, Process Model
2	Requirement Analysis: • Creating a Data Flow • Data Dictionary, Use Cases
3	Project Management: Computing FP Effort Schedule, Risk Table, Timeline chart

4	Design Engineering: • Architectural Design • Data Design, Component Level Design
5	Testing:
	• Basis Path Testing

Sample Projects:

1. **Criminal Record Management**: Implement a criminal record management system for jailers, police officers and CBI officers
2. **DTC Route Information**: Online information about the bus routes and their frequency and
fares
3. **Car Pooling**: To maintain a web based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.
4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation system
8. Automatic Internal Assessment System
9. Parking Allocation System
10. Wholesale Management System

COSMJ503: Database Management Systems Theory

LIST OF COURSE OUTCOMES

CO1 Students will be able to Discuss the fundamental concepts of relational database and SQL
CO2 Students will be able to Use ER model for Relational model mapping to perform database design
effectively
CO3 Students will be able to Summarize the properties of transactions and concurrency control mechanisms
CO4 Students will be able to Outline the various storage and optimization techniques
CO5 Students will be able to Compare and contrast various indexing strategies in different database systems and to explain the different advanced database

1.Introduction

Characteristics of database approach, data models, database system architecture and data independence.

2.Entity Relationship(ER) Modeling **(8 Lectures)**

Entity types, relationships, constraints.

3.Relation data model **(20 Lectures)**

Relational model concepts, relational constraints, relational algebra, SQL queries

4.Database design

Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition, Normal forms(upto BCNF).

5.Transaction Processing (3 Lectures)

ACID properties, concurrency control

6.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.

Books Recommended:

1.R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.

2.R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.

Database Management Systems Lab

LIST OF COURSE OUTCOMES

CO1 Students will be able to Use typical data definitions and manipulation commands.
CO2 Students will be able to Design applications to test Nested and Join Queries
CO3 Students will be able to Implement simple applications that use Views
CO4 Students will be able to Implement applications that require a Front-end Tool
CO5 Students will be able to Critically analyze the use of Tables, Views, Functions and Procedures

1. Structured Query Language

Creating Database
Creating a Table
Specifying Relational Data Types
Specifying Constraints
Creating Indexes

2. Table and Record Handling

INSERT statement
Using SELECT and INSERT together
DELETE, UPDATE, TRUNCATE statements
DROP, ALTER statements

3. Retrieving Data from a Database

The SELECT statement
Using the WHERE clause
Using Logical Operators in the WHERE clause
Using IN, BETWEEN, LIKE , ORDER BY, GROUP BY and HAVING Clause
Using Aggregate Functions
Combining Tables Using JOINS
Subqueries

4. Database Management

Creating Views
Creating Column Aliases
Creating Database Users
Using GRANT and REVOKE

COSMJ504: Discrete Structures Theory:

60 Lectures

LIST OF COURSE OUTCOMES

CO1 Students will be able to Gain the knowledge of the concepts needed to test the logic of a program
CO2 Students will be able to Explaining The basics of the counting principles
CO3 Students will be able to Explaining the basic concepts of combinatory and graph theory
CO4 Students will be able to Describe the concepts and properties of algebraic structures such as groups, rings
and fields
CO5 Students will be able to Explaining the concepts and significance of lattices and Boolean algebra which are widely used in computer science and engineering

1.Introduction: (15 Lectures)

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.

2.Growth of Functions: **(8 Lectures)**

Asymptotic Notations, Summation formulas and properties, Bounding Summations, approximation by Integrals

3.Recurrences: (10 Lectures)

Recurrence Relations, generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Recurrence Trees, Master Theorem

4.Graph Theory (15 Lectures)

Basic Terminology, Models and Types, multigraphs and weighted graphs, Graph Representaion, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring, Trees, Basic Terminology and properties of Trees, Introduction to Spanning Trees

5.Prepositional Logic (12 Lectures)

Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory

Recommended Books:

1.C.L. Liu , D.P. Mahopatra, Elements of Discrete mathematics, 2nd Edition , Tata McGraw Hill, 1985,

1. Kenneth Rosen, Discrete Mathematics and Its Applications, Sixth Edition ,McGraw Hill 2006

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

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

J. L. Hein, Discrete Structures, Logic, and Computability, 3rd Edition, Jones and Bartlett Publishers, 2009

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

COSMJ601: Machine Learning:

(40 Lecture)

LIST OF COURSE OUTCOMES

CO1 Learn the basics of learning problems with hypothesis and version spaces
CO2 Understand the features of machine learning to apply on real world problems
CO3 Characterize the machine learning algorithms as supervised learning and unsupervised learning and Apply and
analyzethe various algorithms of supervised and unsupervised learning
CO4 Analyze the concept of neural networks for learning linear and non-linear activation functions
CO5 Learn the concepts in Bayesian analysis from probability models and methods
CO6 Understand the fundamental concepts of Genetic Algorithm and Analyze and design the genetic algorithms for
optimization engineering problems

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.
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.
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.
Regularization: Regularization and its utility: The problem of Overfitting, Application of Regularization in Linear and Logistic Regression, Regularization and Bias/Variance.
Neural Networks: Introduction, Model Representation, Gradient Descent vs. Perceptron Training, Stochastic Gradient Descent, Multilayer Perceptrons, Multiclass Representation, Back propagation Algorithm.

Suggested Books:

1. Ethem Alpaydin, "Introduction to Machine Learning" 2nd Edition, The MIT Press, 2009.
2. Tom M. Mitchell, "Machine Learning", First Edition by Tata McGraw-Hill Education, 2013.
3. Christopher M. Bishop, "Pattern Recognition and Machine Learning" by Springer, 2007.
4. Mevin P. Murphy, "Machine Learning: A Probabilistic Perspective" by The MIT Press, 2012.

Machine Learning Lab : **20 Lectures**

LIST OF COURSE OUTCOMES

For practical Labs for Machine Learning, students may use softwares like MABLAB/Octave or Python. For later exercises, students can create/use their own datasets or utilize datasets from online repositories like UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/).

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, additing/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 14.

COSMJ602: Theory of Computation **Theory: 40 Lectures**

LIST OF COURSE OUTCOMES

CO1 Students will be able to Construct automata, regular expression for any pattern.
CO2 Students will be able to Write Context free grammar for any construct
CO3 Students will be able to Design Turing machines for any language.
CO4 Students will be able to Propose computation solutions using Turing machines.
CO5 Students will be able to Derive whether a problem is decidable or not

1. Languages (8 Lectures)

Alphabets, string, language, Basic Operations on language, Concatenation, KleeneStar

2. Finite Automata and Regular Languages (20 Lectures)

Regular Expressions, Transition Graphs, Deterministics and non-deterministic finite automata, NFA to DFA Conversion, Regular languages and their relationship with finite automata, Pumping lemma and closure properties of regular languages.

3. Context free languages (17 Lectures)

Context free grammars, parse trees, ambiguities in grammars and languages, Pushdown automata (Deterministic and Non-deterministic), Pumping Lemma, Properties of context free languages, normal forms.

4.Turing Macines and Models of Computations (15 Lectures)

RAM, Turing Machine as a model of computation, Universal Turing Machine, Language acceptability, decidability, halting problem, Recursively enumerable and recursive languages, unsolvability problems.

Recommended Books:

1. Daniel I.A.Cohen, Introduction to computer theory, John Wiley,1996
2. Lewis & Papadimitriou, Elements of the theory of computation , PHI 1997.

3. Hoperoft, Aho, Ullman, Introduction to Automata theory, Language & Computation **–3rd** Edition, Pearson Education. 2006

4. P. Linz, An Introduction to Formal Language and Automata 4th Bartlett, 2006

COSMJ603: Computer Graphics :

40 Lectures

LIST OF COURSE OUTCOMES

CO1 Understand the basics of computer graphics, different graphics systems and applications of computer graphics.
CO2 Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis.
CO3 Use of geometric transformations on graphics objects and their application in composite form.
CO4 Extract scene with different clipping methods and its transformation to graphics display device.
CO5 Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.
CO6 Render projected objects to naturalize the scene in 2D view and use of illumination models for this.

1.Introduction **(5 Lectures)**

Basic elements of Computer graphics, Applications of Computer Graphics.

2.Graphics Hardware **(8 Lectures)**

Architecture of Raster and Random scan display devices, input/output devices.

3.Fundamental Techniques in Graphics **(22 Lectures)**

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.

4.Geometric Modeling **(10 Lectures)**

Representing curves & Surfaces.

5.Visible Surface determination **(8 Lectures)**

Hidden surface elimination.

6.Surface rendering **(7 Lectures)**

Illumination and shading models. Basic color models and Computer Animation.

Books Recommended:

1.J.D.Foley, A.Van Dan, Feiner, Hughes Computer Graphics Principles & Practice 2nd edition Publication Addison Wesley 1990.

2.D.Hearn, Baker: Computer Graphics, Prentice Hall of India 2008.

3.D.F.Rogers Procedural Elements for Computer Graphics, McGraw Hill 1997.

4.D.F.Rogers, Adams Mathematical Elements for Computer Graphics, McGraw Hill 2nd edition 1989.

Computer Graphics Lab: **20 Lectures**

LIST OF COURSE OUTCOMES

CO1 Using OpenGL for Graphics
CO2 Programming User-interface issues
CO3 Concepts of 2D & 3D object representation
CO4 Implementation of various scan & clipping algorithms
CO5 2D modeling
CO6 Implementation of illumination model for rendering 3D objects
CO7 Visibility detection & 3D viewing
CO8 Implementation of a project based on learned concepts

.

1.Write a program to implement Bresenham‘s line drawing algorithm.
2.Write a program to implement mid-point circle drawing 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.

COSMJ604:

PROJECT

SKILL ENHANCEMENT COURSE

1.HTML Programming
2.Introduction to python programming.
3. MATLAB Proframming.

COSSEC01:Web Designing

Course Outcome

After completing this course, students will be able to structure, organize, and publish a website using HTML5 and CSS including how to view a site's HTML and understand its structure, purpose.

- **Unit-I: Introduction**
- **Unit-II: The Basics**
  - The Head, the Body
  - Colors, Attributes
  - Lists, ordered and unordered
- **Unit-III: Links**
  - Introduction
  - Relative Links, Absolute Links
  - Link Attributes
  - Using the ID Attribute to Link Within a Document
- **Unit-IV: Images**
  - Putting an Image on a Page
  - Using Images as Links
  - Putting an Image in the Background
- **Unit V: – Tables**
  - Creating a Table
  - Table Headers
  - Captions
  - Spanning Multiple Columns
  - Styling Table
- **Unit VI – Forms**
  - Basic Input and Attributes
  - Other Kinds of Inputs
  - Styling forms with CSS
  - Where To Go From Here

Unit –V

Introduction to Cascading Style Sheets (CSS)

The style tag
Tag selectors
The font-size, font-family, color, & line-height properties
Hexadecimal color codes

CSS Class Selectors

- The class attribute
- CSS class selectors
- The span tag
- CSS opacity

Div Tags, ID Selectors, & Basic Page Formatting

- Dividing up content with the div tag
- Assigning IDs to divs
- Setting width & max-width
- CSS background-color
- Adding padding inside a div
- Centering content
- CSS borders
- CSS shorthand & the DRY principle

Using Browser Developer Tools

- Opening the DevTools in Chrome
- Editing HTML in the DevTools Elements panel
- Enabling, disabling, & editing CSS in the DevTools
- Using DevTools to fine-tune your CSS
- Hexadecimal shorthand

HTML5 Semantic Elements & Validating HTML

- The outline algorithm
- The header, nav, aside, & footer elements
- Understanding articles & sections
- The main element
- The figure & figcaption elements
- Checking for errors: validating your code

Unit-VII

Basic Concept of Node Js

Book Recommended:

1. Virginia DeBolt , Integrated HTML and CSS A Smarter, Faster Way to Learn Wiley / Sybex , 2006
2. Cassidy Williams, Camryn Williams Introduction to **HTML** and CSS, O'Reilly, 2015

COSSEC02: Computing Paradigm with Python Programming

Course outcome:

1.Write, Test and Debug Python Programs
2.Use functions and represent Compound data using Lists, Tuples and Dictionaries
3.Read and write data from & to files in Python and develop Application using Pygame

Brief introduction: What is Python, Uses of Python Programming Language / Python Applications, Features of Python Programming Language, Implementations of Python, and Python career opportunities.

Python Comments: Purpose/use of comments in Computer Programming, Comments for Understanding Python code, Python Comment Syntax, Python Single line comment, Multiline comment in Python, and writing Python comments.

Python Keywords and Identifiers: Python keywords or Reserved words, The syntax and structure of the Python language, Python keywords are case sensitive, Python literals (True, False, Null), Python Identifiers, class names, variable names, function names, method names, and Identifier naming rules.

Python Variables and Numbers: What is Variable?, Declaration of Variables, Assign Values to Variables, Initialization, Reading, Variable naming restrictions, and Types of Python Variables, Numbers.

Python Data Types: What is Data Type?, Implicit Declaration of Data Types, Python Numbers (Integers, floating-point numbers, and complex numbers), Python Strings, Python boolean data type.

Python Operators: Python Arithmetic, Comparison/Relational Operators, Increment Operators, Logical operators, Python Identity Operators, and Python Operators Precedence.

2. Python Control Flow – Decision Making and Looping

Decision Making in Python: Indentation, Simple If Structure, if-else structure, if elif structure, and nested If Structure.

Python Loop Statements:Python while loop, Python for loop, Python range(), Python Nested Loop Structures, and Inserting conditions in Loops and vice versa.

PythonBranching Statements:break, continue.

3. Python User Input/Output

Python user input from the keyboard can be read using the input() built-in function. The input from the user is read as a string and can be assigned to a variable. Python output

statement.

4. Python Strings, List, Tuples, Set, Dictionaries String:Finding String length, Concatenating Strings, Print a String multiple times, Check whether the String has all numeric characters and Check whether the String has all alphabetic characters?.

Lists: Create Python Lists, Display a list, Update Python Lists, and Add an element into Python List, Delete Elements from Python Lists, Sorting a list of Strings, Searching the List, and Built-in Functions & Built-in Methods for Python Lists.

Tuples: Some features, Conversion, Differences between tuples and lists.

Sets: Methods, Conversion, Set Theoretic Operations.

Dictionaries: Key and values, Attributes, Creating a Dictionary.

5. Python user-defined and Built-in Functions

Define user-defined function,Severalbuilt-in functions in Python.

6. Python Scripts for elementary Computations

Roots of quadratic Equation, Statistical Calculations, Factorial, Combination, Conversion of number system, Armstrong number, Strong number, Prime number, Palindrome number, Fibonacci number, Infinite series, Random number.

9. Numpy

Introducing Arrays: What is an array? Shape of an array, Dimensions, Resize, Creating one dimensional arrays, Algebra with arrays, Some useful functions, Array as Matrix, Matrix Operations, Arrays as Vectors, Polynomial by Numpy. Curve Fitting by Numpy:Fit with a user defined function.

Python Programming Lab (COSSEC01):

Using for loop, print a table of Celsius/Fahrenheit equivalences. Let c be the Celsius temperatures ranging from 0 to 100, for each value of c, print the corresponding Fahrenheit temperature.

1. Using while loop, produce a table of sins, cosines and tangents. Make a variable x in range from 0 to 10 in steps of 0.2. For each value of x, print the value of sin(x), cos(x) andtan(x).

2. Write a program that reads an integer value and prints —leap year‖ or —not a leapyear‖.

3. Write a program that takes a positive integer n and then produces n lines of output shown as follows.
   For example enter a size: 5
   *
   **
   ***
   ****
   *****

4. Write a function that takes an integer ‗n' as input and

calculates the value of 1 + 1/1! + 1/2! + 1/3! + … +1/n

5. Write a function that takes an integer input and calculates the factorial of thatnumber.
6. Write a function that takes a string input and checks if it's a palindrome ornot.
7. Write a list function to convert a string into a list, as in list (‗abc') gives [a, b,c].
8. Write a program to generate Fibonacciseries.
9. Write a program to check whether the input number is even orodd.
10. Write a program to compare three numbers and print the largestone.
11. Write a program to print factors of a given number.
12. Write a method to calculate GCD of twonumbers.
13. Write a program to create Stack Class and implement all its methods. (UseLists).
14. Write a program to create Queue Class and implement all its methods. (UseLists)
15. Write a program to implement linear and binary search onlists.
16. Write a program to sort a list using insertion sort and bubble sort and selectionsort.

SEC03: Programming in MATLAB

LIST OF COURSE OUTCOMES

Unit I- Introduction to Programming: Components of a computer, working with numbers, Machine code, Software hierarchy. (2L)
Unit II- Programming Environment: MATLAB Windows, A First Program, Expressions, Constants,Variables and assignment statement, Arrays. (3L)
Unit III- Graph Plots: Basic plotting, Built in functions, Generating waveforms, Sound replay, load and save. (2L)
Unit IV- Procedures and Functions: Arguments and return values, M-files, Formatted console input-output, String handling. (3L)
Unit V-Control Statements: Conditional statements: If, Else, Else-if, Repetition statements: While,for loop. (2L)
Unit VI- Manipulating Text: Writing to a text file, Reading from a text file, Randomising and sorting a list, searching a list. (2L)
Unit VII- GUI Interface: Attaching buttons to actions, Getting Input, Setting Output. (2L)

Recommended Books:

1.MATLAB: An Introduction with Applications, by Amos Gilat, 2nd edition, Wiley, 2004,
2.C.B. Moler, Numerical Computing with MATLAB, SIAM, 2004

Software Lab Based on MatLab:

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. $(0.0000123 + 5.67\times10^{-3}) \times 0.4567\times10^{-4}$

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,…,√Nth entries, i.e. those numbers which have indices that are square numbers.
8. Draw a graph that joins the points (0,1), (4,3), (2,0) and (5,-2).
9. The seeds on a sunflower are distributed according to the formula below. Plot a small circle at each of the first 1000 co-ordinates :

$$r_n = \sqrt{n}$$

$$\theta_n = \frac{137.51}{180}\pi n$$

10. Calculate 10 approximate points from the function y=2x by using the formulae:
    i. $x_n = n$
    ii. $y_n = 2n + \text{rand} - 0.5$

    Fit a line of best fit to these points using the function polyfit() with degree=1, and generate co-ordinates from the line of best fit using polyval(). Use the on-line help to find out how to use these functions. Plot the raw data and the line of best fit.
11. Calculate and replay 1 second of a sinewave at 500Hz with a sampling rate of 11025Hz. Save the sound to a file called "ex35.wav". Plot the first 100 samples.
12. Calculate and replay a 2 second chirp. That is, a sinusoid that steadily increases in frequency with time, from say 250Hz at the start to 1000Hz at the end.
13. Build a square wave by adding together 10 odd harmonics: 1f, 3f, 5f, etc. The amplitude of the nth harmonic should be 1/n. Display a graph of one cycle of the result superimposed on the individual harmonics.
14. Write a function called FtoC (ftoc.m) to convert Fahrenheit temperatures into Celsius. Make sure the program has a title comment and a help page. Test from the command window with:

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

15. Write a program to input 2 strings from the user and to print out (i) the concatenation of the two strings with a space between them, (ii) a line of asterisks the same length as the concatenated strings, and (iii) the reversed concatenation. For example:
    i. Enter string 1: Mark
    ii. Enter string 2: Huckvale
    iii. Mark Huckvale
    iv. **********
    v. elavkcuH kraM