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

Syllabus For
B.Sc. (Honours) Computer Science
Choice Based Credit System
(CBCS)

1st Semester

Course	Course Type/ Nature	Course Code	Name of the Subject	Teaching Scheme in hour per week			Credit	Marks
				L	T	P		
CC-1	Core Course - 1	C1T	Programming Fundamentals usingC/C++	4	0	0	4	75
		C1P	Programming Fundamentals usingC/C++ LAB	0	0	4	2	
CC-2	Core Course - 2	C2T	Computer System Architecture	4	0	0	4	75
		C2P	Computer System Architecture LAB	0	0	4	2	
GE-1	Generic Elective -1	GE1T	Computer Fundamentals	4	0	0	4	75
		GE1P	Computer Fundamentals LAB	0	0	2	2	
AEC-1	Ability Enhancement Compulsory Course-1		English	1	0	2	2	50
1st Semester Total							20	275

2nd Semester

Course	Course Type/ Nature	Course Code	Name of the Subject	Teaching Scheme in hour per week			Credit	Marks
				L	T	P		
CC-3	Core Course – 3	C3T	Programming in JAVA	4	0	0	4	75
		C3P	Programming in JAVA LAB	0	0	4	2	
CC-4	Core Course - 4	C4T	Data Structures	4	0	0	4	75
		C4P	Data Structures LAB	0	0	4	2	
GE-2	Generic Elective -2	GE2T	Introduction to Programming	4	0	0	4	75
		GE2P	Introduction to Programming LAB	0	0	2	2	
AEC-2	Ability Enhancement Compulsory Course-2		Environmental Science	1	0	2	2	50
2nd Semester Total							20	275

3rd Semester

Course	Course Type/ Nature	Course Code	Name of the Subject	Teaching Scheme in hour per week			Credit	Marks
				L	T	P		
CC-5	Core Course - 5	C5T	Discrete Structures	4	0	0	5	75
		C5P	Discrete Structures Tutorial	0	1	0	1	
CC-6	Core Course - 6	C6T	Operating Systems	4	0	0	4	75
		C6P	Operating Systems LAB	0	0	4	2	
CC-7	Core Course - 7	C7T	Computer Networks	4	0	0	4	75
		C7P	Computer Networks LAB	0	0	4	2	
GE-3	Generic Elective -3	GE3T	Computer Fundamentals and Python Programming	4	0	0	4	75
		GE3P	Python Programming LAB	0	0	2	2	
SEC-1	Skill Enhancement Course-1		SEC-1	1	0	2	2	50
3rd Semester Total							26	350

4th Semester

Course	Course Type/ Nature	Course Code	Name of the Subject	Teaching Scheme in hour per week			Credit	Marks
				L	T	P		
CC-8	Core Course - 8	C8T	Design and Analysis of Algorithms	4	0	0	6	75
		C8P	Design and Analysis of Algorithms LAB	0	0	4		
CC-9	Core Course - 9	C9T	Software Engineering	4	0	0	6	75
		C9P	Software Engineering LAB	0	0	4		
CC-10	Core Course - 10	C10T	Database Management System	4	0	0	6	75
		C10P	Database Management System LAB	0	0	4		
GE-4	Generic Elective -4	GE4T	Data Structure	4	0	0	4	75
		GE4P	Data Structure LAB	0	0	2	2	
SEC-2	Skill Enhancement Course-2		SEC-2	1	0	2	2	50
4th Semester Total							26	350

5th Semester

Course	Course Type/ Nature	Course Code	Name of the Subject	Teaching Scheme in hour per week			Credit	Marks
				L	T	P		
CC-11	Core Course – 11	C11T	Internet Technologies	4	0	0	4	75
		C11P	Internet Technologies LAB	0	0	4	2	
CC-12	Core Course - 12	C12T	Theory of Computation	4	0	0	5	75
		C12P	Theory of Computation Tutorial	0	4	0	1	
DSE-1	Discipline Specific Elective-1		DSE-1	4	0	0	4	75
			DSE-1 Lab	0	0	4	2	
DSE-2	Discipline Specific Elective-2		DSE-2	4	0	0	4	75
			DSE-2 Lab	0	0	2	2	
5th Semester Total							24	300

6th Semester

Course	Course Type/ Nature	Course Code	Name of the Subject	Teaching Scheme in hour per week			Credit	Marks
				L	T	P		
CC-13	Core Course – 13	C13T	Artificial Intelligence	4	0	0	4	75
		C13P	Artificial Intelligence LAB	0	0	4	2	
CC-14	Core Course – 14	C14T	Computer Graphics	4	0	0	4	75
		C14P	Computer Graphics LAB	0	0	4	2	
DSE-3	Discipline Specific Elective-3		DSE-3	4	0	0	4	75
			DSE-3 Lab	0	0	2	2	
DSE-4	Discipline Specific Elective-4		Project Work / Dissertation	0	0	4	4	75
			Practical/Tutorial	0	0	2	2	
6th Semester Total							24	300

Skill Enhancement Courses (Credit: 02 each) – SEC1 (ANY ONE)

1. Android Programming (1) + Lab (2)
2. Programming in MATLAB (1) + Lab (2)
3. HTML Programming (1) + Lab (2)

SEC 2 (ANY ONE)

6. Programming in Python (1) + Lab(2)
7. PHP Programming (1) + Lab (2)
8. UNIX/LINUX Programming (1) + Lab (2)
9. R Programming (1) + Lab (2)

Discipline Specific Elective Papers: (Credit: 06 each) (3 papers to be selected) – DSE 1

1. Information Security (4) + Lab (4)
3. Microprocessor (4) + Lab (4)

DSE 2

4. Computational Linguistics (4) + Lab (4)
5. Digital Image Processing (4) + Lab (4)
6. Machine Learning (4) + Lab (4)

DSE 3

7. Introduction to Data Sciences (4) + Lab (4)
8. Cloud Computing (4) + Lab (4)
9. Numerical Methods (4) + Lab (4)
10. System Programming (4) + Lab (4)
12. Data Mining (4) + Lab(4)
13. Big Data Analytics (4) + Lab(4)
14. Soft Computing (4) + Lab (4)

CORE COURSES (HONOURS) IN COMPUTER SCIENCE

COMPUTER SCIENCE (C-I): Programming Fundamentals using C/C++

Theory: 60 Lectures

1. Introduction to C and C++ (3 Lectures)

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

2. Data Types, Variables, Constants, Operators and Basic I/O (5 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, putcharetc), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.hetc).

3. Expressions, Conditional Statements and Iterative Statements (5 Lectures)

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

4. Functions and Arrays (10 Lectures)

Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments.

Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays

5. Derived Data Types (Structures and Unions) (3 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 data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.

6. Pointers and References in C++ (7 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. Pointers vs. References, Declaring and initializing references, Using references as function arguments and function return values

7. Memory Allocation in C++ (3 Lectures)

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

8. File I/O, Preprocessor Directives (4 Lectures)

Opening and closing a file (use of fstream header file, ifstream, ofstream and fstream classes), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros

9. Using Classes in C++ (7 Lectures)

Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Function overloading in classes, Class Variables &Functions, Objects as parameters, Specifying the Protected and Private Access, Copy Constructors,Overview of Template classes and their use.

10. Overview of Function Overloading and Operator Overloading (5 Lectures)

Need of Overloading functions and operators, Overloading functions by number and type of arguments, Looking at an operator as a function call, Overloading Operators (including assignment operators, unary operators)

11. Inheritance, Polymorphism and Exception Handling (8 Lectures)

Introduction to Inheritance (Multi-Level Inheritance, Multiple Inheritance), Polymorphism (Virtual Functions, Pure Virtual Functions), Basics Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions, Restricting exceptions, Rethrowing exceptions.

Reference Books

1. HerbtzSchildt, "C++: The Complete Reference", Fourth Edition, McGraw Hill.2003

2. BjarneStroustrup, "The C++ Programming Language", 4th Edition, Addison-Wesley , 2013.

3. BjarneStroustroup, "Programming -- Principles and Practice using C++", 2nd Edition, Addison-Wesley 2014.

4. E Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw-Hill Education, 2008.

5. Paul Deitel, Harvey Deitel, "C++ How to Program", 8th Edition, Prentice Hall, 2011.

5. John R. Hubbard, "Programming with C++", Schaum's Series, 2nd Edition, 2000.

6. Andrew Koeni, Barbara, E. Moo, "Accelerated C++", Published by Addison-Wesley , 2000.

7. Scott Meyers, "Effective C++", 3rd Edition, Published by Addison-Wesley, 2005.

8. Harry, H. Chaudhary, "Head First C++ Programming: The Definitive Beginner's Guide", First Create space Inc, O-D Publishing, LLC USA.2014

9. Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.

10. Stanley B. Lippman, JoseeLajoie, Barbara E. Moo, "C++ Primer", Published by Addison-Wesley, 5th Edition, 2012

COMPUTER SCIENCE LAB (C-I): Programming Fundamentals using C/C++ Lab
(Using GCC/Visual C++ compiler)
Practical: 60 Lectures

---

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.

COMPUTER SCIENCE (C-II): Computer System Architecture

Theory: 60 Lectures

1. Introduction (8 lectures)

Logic gates, boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexers, registers, counters and memory units.

2. Data Representation and Basic Computer Arithmetic (10 lectures)

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

3. Basic Computer Organization and Design (13 lectures)

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

4. Central Processing Unit (15 lectures)

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

5. Memory Organization (6 lectures)

Cache memory, Associative memory, mapping.

6. Input-Output Organization (8 lectures)

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

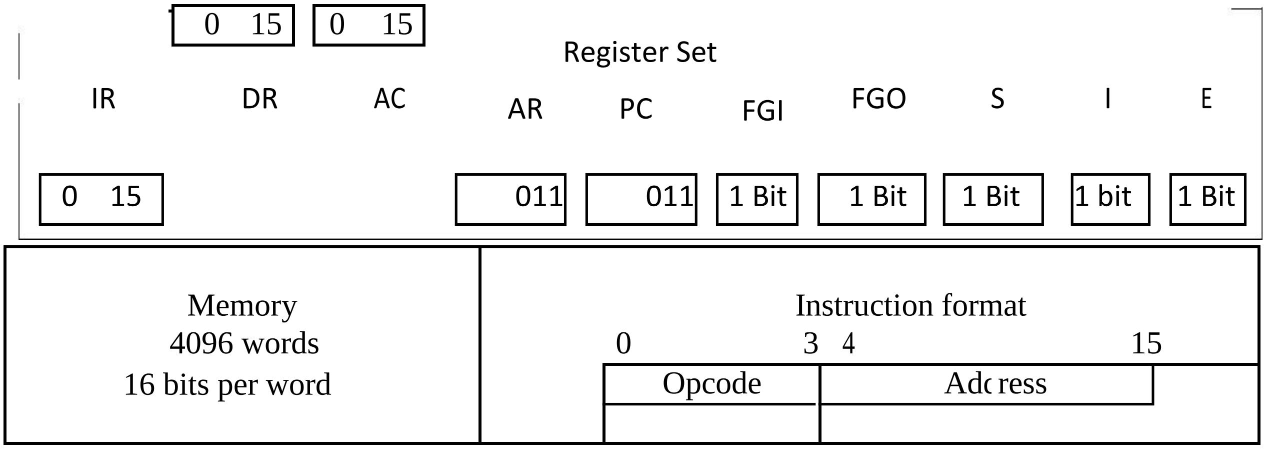
Recommended Books:

1. M. Mano, Computer System Architecture, Pearson Education 1992
2. A. J. Dos Reis, Assembly Language and Computer Architecture using C++ and JAVA, Course Technology, 2004
3. W. Stallings, Computer Organization and Architecture Designing for Performance, 8th Edition, Prentice Hall of India,2009
4. M.M. Mano , Digital Design, Pearson Education Asia,2013
5. Carl Hamacher, Computer Organization, Fifth edition, McGrawHill, 2012.

COMPUTER SCIENCE LAB (C-II): Computer System Architecture
Lab Practical: 60 Lectures

(Practical exercises based on concepts listed in theory)

1. Create a machine based on the following



Basic Computer Instructions

Memory Reference			Register Reference		Input-Output	
Symbol	Hex		Symbol	Hex	Symbol	Hex
AND	0xxx	Direct Addressing	CLA	E800	INP	F800
ADD	2xxx		CLE	E400	OUT	F400
LDA	4xxx		CMA	E200	SKI	F200
STA	6xxx		CME	E100	SKO	F100
BUN	8xxx		CIR	E080	ION	F080
BSA	Axxx		CIL	E040	IOF	F040
ISZ	Cxxx		INC	E020		
AND_I	1xxx	Indirect Addressing	SPA	E010		
ADD_I	3xxx		SNA	E008		
LDA_I	5xxx		SZA	E004		
STA_I	7xxx		SZE	E002		
BUN_I	9xxx		HLT	E001		
BSA_I	Bxxx					
ISZ_I	Dxxx					

Optional

Refer to Chapter-5 of Morris Mano for description of instructions.

2. Create the micro operations and associate with instructions as given in the chapter (except interrupts). Design the register set, memory and the instruction set. Use this machine for the assignments of this section.

3. Create a Fetch routine of the instruction cycle.

4. Simulate the machine to determine the contents of AC, E, PC, AR and IR registers in hexadecimal after the execution of each of following register reference instructions:

   a. CLA
   b. CLE
   c. CMA
   d. CME
   e. CIR
   f. CIL
   g. INC
   h. SPA
   i. SNA
   j. SZA
   k. SZE
   l. HLT

   Initialize the contents of AC to (A937)$_{16}$, that of PC to (022)$_{16}$ and E to 1.

5. Simulate the machine for the following memory-reference instructions with I= 0 and address part = 082. The instruction to be stored at address 022 in RAM. Initialize the memory word at address 082 with the operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

   a. ADD
   b. AND
   c. LDA
   d. STA
   e. BUN
   f. BSA
   g. ISZ

6. Simulate the machine for the memory-reference instructions referred in above question with I= 1 and address part = 082. The instruction to be stored at address 026 in RAM. Initialize the memory word at address 082 with the value 298. Initialize the memory word at address 298 with operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

7. Modify the machine created in Practical 1 according to the following instruction format:

Instruction format

0 2	3	4 15
Opcode	I	Address

   a. The instruction format contains a 3-bit opcode, a 1-bit addressing mode and a 12-bit address. There are only two addressing modes, I = 0 (direct addressing) and I = 1 (indirect addressing).

   b. Create a new register I of 1 bit.

   c. Create two new microinstructions as follows :

      i. Check the opcode of instruction to determine type of instruction (Memory Reference/Register Reference/Input-Output) and then jump accordingly.
      ii. Check the I bit to determine the addressing mode and then jump accordingly.

COMPUTER SCIENCE (C-III): Programming in Java

Theory: 60 Lectures

1. 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. Object-Oriented Programming Overview (4 Lectures)

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.

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)

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.

5. 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.

COMPUTER SCIENCE LAB (C-III): Programming in Java Lab

Practical: 60 Lectures

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 StringBufferclasss 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.

COMPUTER SCIENCE (C-V): Data Structures

Theory: 60 Lectures

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** **(10 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** **(5 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** **(20 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.

COMPUTER SCIENCE LAB (C-V): Data Structures Lab

Practical: 60 Lectures

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.

COMPUTER SCIENCE (C-V): Discrete Structures Theory: 60 Lectures

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,
2. 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
5. 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

COMPUTER SCIENCE LAB (C-V): Discrete Structures Tutorial

Tutorial: 15 lectures

COMPUTER SCIENCE (C-VI): Operating Systems

Theory: 60 Lectures

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 systemprograms.

3. **Process Management** (**20**Lectures)
System view of the process and resources, process abstraction, processhierarchy, 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 -fixedand variable partitions,

paging, segmentation, virtual memory

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

Directory structure, file operations, file allocation methods, devicemanagement.

6. Protection and Security (4 Lectures)

Policy mechanism, Authentication, Internal access Authorization.

Recommended Books:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
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 , 5th Edition, Prentice Hall of India. 2008.
5. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

COMPUTER SCIENCE LAB (C-VI): Operating Systems Lab

Practical: 60 Lectures

C/ C++ programs

1. WRITE A PROGRAM (using *fork()* and/or *exec()* commands) where parent and child execute:
   a) same program, same code.
   b) same program, different code.
   c) before terminating, the parent waits for the child to finish its task.

2. WRITE A PROGRAM to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information)
3. WRITE A PROGRAM to report behaviour of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
4. WRITE A PROGRAM to print file details including owner access permissions, file access time, where file name is given as argument.
5. WRITE A PROGRAM to copy files using system calls.
6. Write program to implement FCFS scheduling algorithm.
7. Write program to implement Round Robin scheduling algorithm.
8. Write program to implement SJF scheduling algorithm.
9. Write program to implement non-preemptive priority based scheduling algorithm.
10. Write program to implement preemptive priority based scheduling algorithm.
11. Write program to implement SRJF scheduling algorithm.
12. Write program to calculate sum of n numbers using *thread* library.
13. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

COMPUTER SCIENCE (C-VII): Computer Networks Theory: 60 Lectures

1. Introduction to Computer Networks (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-ratelimits; 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 mechanisms (10 Lectures)

Circuit switching; packetswitching- 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; connectingLAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways;

6. Networks Layer Functions and Protocols (6 Lectures)

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

7. Transport Layer Functions and Protocols (6 Lectures)

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

8. 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 SCIENCE LAB (C-VII): Computer Networks Lab

Practical: 60 Lectures

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

COMPUTER SCIENCE (C-VIII): Design and Analysis of Algorithms

Theory: 60 Lectures

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, DynamicProgramming, Greedy Algorithms.

3. Sorting and Searching Techniques (20 Lectures

Elementary sorting techniques–Bubble Sort, InsertionSort, 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;

34.Lower Bounding Techniques (5 Lectures)

Decision Trees

5.Balanced Trees (7 Lectures)

Red-Black Trees

6.Advanced Analysis Technique (5 Lectures)

Amortized analysis

7.Graphs (5 Lectures)

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

8. String Processing (5Lectures)

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

COMPUTER SCIENCE LAB (C-VIII): Design and Analysis of Algorithms Lab
Practical: 60 Lectures

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.

COMPUTER SCIENCE (C-IX): Software Engineering Theory: 60 Lectures

1.Introduction (8 Lectures)

The Evolving Role of Software, Software Characteristics, Changing Nature ofSoftware, Software Engineering as a Layered Technology, Software Process Framework, Framework and Umbrella

2.Requirement Analysis (10 Lectures)

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

3.Software Project Management (8Lectures)

Estimation in Project Planning Process, Project Scheduling.

Software Risks, 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 SoftwareTesting, 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), McGraw-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.

COMPUTER SCIENCE LAB (C-IX): Software Engineering Lab
Practical: 60 Lectures

S. No	Practical Title
1.	• Problem Statement, • Process Model
2.	Requirement Analysis: • Creating a Data Flow • Data Dictionary, Use Cases Project Management:
3.	• 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

COMPUTER SCIENCE (C-X): Database Management Systems Theory: 60 Lectures

1.Introduction (6 Lectures)

Characteristics of database approach, data models, database system architectureand 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, SQLqueries

4.Database design (15 Lectures)

Mapping ER/EER model to relational database, functional dependencies,Lossless decomposition, Normalforms(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 f+iles(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. th

3. A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts 6 Edition, McGraw Hill, 2010.
4. R. Elmasri, S.B$_{t \cdot h}$Navathe Database Systems Models, Languages, Design and application Programming, 6 Edition, Pearson Education,2013.

COMPUTER SCIENCE LAB (C-X): Database Management Systems Lab Practical: 60 Lectures

---

Structured Query Language

1. 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

COMPUTER SCIENCE (C-XI): Internet Technologies Theory: 60 Lectures

Java **(5 lectures)**

Use of Objects, Array and ArrayList class

JavaScript **(15 lectures)**

Data types, operators, functions, control structures, events and event handling.

JDBC **(10 lectures)**

JDBC Fundamentals, Establishing Connectivity and working with connection interface, Working with statements, Creating and Executing SQL Statements, Working with Result Set Objects.

JSP **(20 lectures)**

Introduction to JavaServer Pages, HTTP and Servlet Basics, The Problem with Servlets, The Anatomy of a JSP Page, JSP Processing, JSP Application Design with MVC, Setting Up the JSP Environment, Implicit JSP Objects, Conditional Processing, Displaying Values, Using an expression to Set an Attribute, Declaring Variables and Methods, Error Handling and Debugging, Sharing Data Between JSP Pages, Requests, and Users, Database Access.

Java Beans **(10 lectures)**

Java Beans Fundamentals, JAR files, Introspection, Developing a simple Bean, Connecting to DB

Recommended Books:

1. Ivan Bayross, Web Enabled Commercial Application Development Using Html, Dhtml,javascript, Perl Cgi , BPB Publications, 2009.
2. Cay Horstmann, BIG Java, Wiley Publication , 3rd Edition., 2009
3. Herbert Schildt , Java 7, The Complete Reference, , 8th Edition, 2009.
4. Jim Keogh ,The Complete Reference J2EE, TMH, , 2002.
5. O'Reilly , Java Server Pages, Hans Bergsten, Third Edition, 2003.

COMPUTER SCIENCE LAB (C-XI): Internet Technologies Lab

Practical: 60 Lectures

Module 1: Assignment of projects

Create event driven program for following:

1. Print a table of numbers from 5 to 15 and their squares and cubes using alert.
2. Print the largest of three numbers.
3. Find the factorial of a number n.
4. Enter a list of positive numbers terminated by Zero. Find the sum and average of these Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.

COMPUTER SCIENCE (C-XII): Theory of Computation

Theory: 60 Lectures

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 edition Publication Jones Bartlett, 2006

COMPUTER SCIENCE (C-XIII): Artificial Intelligence

Theory: 60 Lectures

1. Introduction (06 Lectures)

Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational Agent approaches to AI, Introduction to Intelligent Agents, their structure, behavior and environment.

2. Problem Solving and Searching Techniques (20 Lectures)

Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.

3. Knowledge Representation (20 Lectures)

Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs.

Programming in Logic (PROLOG)

4. Dealing with Uncertainty and Inconsistencies (08 Lectures)

Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.

5. Understanding Natural Languages (06 Lectures)

Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented Transition Nets.

BOOKS RECOMMENDED:

1. DAN.W. Patterson, Introduction to A.I and Expert Systems – PHI, 2007.

2. R$_n$u$_d$ssell &Norvig, Artificial Intelligence-A Modern Approach, LPE, Pearson Prentice Hall, 2 nd edition, 2005.

3. Rich & Knight, Artificial Intelligence – Tata McGraw Hill, 2nd edition, 1991.

4. W.F. Clocksin and Mellish, Programming in PROLOG, Narosa Publishing House, 3rd edition, 2001.

5. Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education, 3rd edition, 2000.

COMPUTER SCIENCE LAB (C-XIII): Artificial Intelligence Lab

Practical: 60 Lectures

1. Write a prolog program to calculate the sum of two numbers.
2. Write a prolog program to find the maximum of two numbers.
3. Write a prolog program to calculate the factorial of a given number.
4. Write a prolog program to calculate the nth Fibonacci number.
5. Write a prolog program, insert_nth(item, n, into_list, result) that asserts that result is the list into_list with item inserted as the n_th element into every list at all levels.
6. Write a Prolog program to remove the Nth item from a list.
7. Write a Prolog program, remove-nth(Before, After) that asserts the After list is the Before list with the removal of every n_th item from every list at all levels.
8. Write a Prolog program to implement append for two lists.
9. Write a Prolog program to implement palindrome(List).
10. Write a Prolog program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y.
11. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List.
12. Write a Prolog program to implement sumlist(List,Sum) so that Sum is the sum of a given list of numbers List.
13. Write a Prolog program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.
14. Write a Prolog program to implement reverse(List,ReversedList) that reverses lists.

15. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List using cut predicate.

16. Write a Prolog program to implement GCD of two numbers.

17. Write a prolog program that implements Semantic Networks/Frame Structures.

COMPUTER SCIENCE LAB (C-XIV): Computer Graphics

Theory: 60 Lectures

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 SCIENCE LAB (C-XIV): Computer Graphics Lab

Practical: 60 Lectures

1. Write a program to implement Bresenham_s line drawing algorithm.
2. Write a program to implement mid-point circle drawing algorithm.
3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
4. Write a program to clip a polygon using Sutherland Hodgeman algorithm.
5. Write a program to apply various 2D transformations on a 2D object (use homogenous coordinates).
6. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.
7. Write a program to draw Hermite/Bezier curve.

Discipline Specific Elective Papers COMPUTER SCIENCE: (Credit: 06 each) (4 papers to be selected) – DSE 1 - 4

I. Information Security

Theory: 60 Lectures

1. Introduction
Security, Attacks, Computer Criminals, Security Services, Security Mechanisms.

2. 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.

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

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

5. Database Security
Requirements, Reliability, Integrity, Sensitive data, Inference, Multilevel Security.

6. Security in Networks

Threats in Networks, Security Controls, firewalls, Intrusion detection systems, Secure e-mails

7. 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. Pfleeger, S. L. Pfleeger; Security in Computing, Prentice Hall of India, 2006
2. W. Stallings; Network Security Essentials: Applications and Standards, 4/E, 2010

Information Security Lab

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois
2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Perform encryption and decryption of Caesar cipher. Write a script for performing these operations.
4. Perform encryption and decryption of a Rail fence cipher. Write a script for performing these operations.
5. Use nmap/zenmap to analyse a remote machine.
6. Use Burp proxy to capture and modify the message.
7. Demonstrate sending of a protected word document.
8. Demonstrate sending of a digitally signed document.
9. Demonstrate sending of a protected worksheet.
10. Demonstrate use of steganography tools.
11. Demonstrate use of gpg utility for signing and encrypting purposes.

Microprocessor

Theory: 60 Lectures

Microprocessor architecture: Internal architecture, system bus architecture, memory and I/O interfaces.

Microprocessor programming: Register Organization, instruction formats, assembly language programming.

Interfacing: Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, display, timer, interrupt controller, DMA controller, video controllers, communication interfaces.

Recommended Books:

1. Barry B. Brey : The Intel Microprocessors : Architecture, Programming and Interfacing. Pearson Education, Sixth Edition,2009.

2. Walter A Triebel, Avtar Singh; The 8088 and 8086 Microprocessors Programming, Interfacing, Software, Hardware, and Applications. PHI, Fourth Edition 2005.

Microprocessor Lab

Practical: 60 Lectures

ASSEMBLY LANGUAGE PROGRAMMING

1. Write a program for 32-bit binary division and multiplication
2. Write a program for 32-bit BCD addition and subtraction
3. Write a program for Linear search and binary search.
4. Write a program to add and subtract two arrays
5. Write a program for binary to ascii conversion
6. Write a program for ascii to binary conversion

IV. Computational Linguistics

Theory: 60 Lectures

Introduction Computers in linguistics and Natural Language Processing The nature and use of text corpora

Introduction to python programming and NLTK, Regular expressions Pattern matching Corpus search and counting

Regular languages Finite-state automata Operations and closure properties Pumping Lemma, Finite-state linguistics Transducers Morphological analysis

N-grams Language modeling Smoothing Evaluation

Part-of-Speech Tagging, Word classes and tagsets Rule-based and stochastic POS tagging Hidden Markov Models Evaluation

Word meaning Semantic ambiguity Semantic relations Semantic roles

Computational lexical semantics, (Un)supervised word sense disambiguation Classifiers Vector-space semantics

References:

1. Jurafsky, D. and J. H. Martin, Speech and language processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (2nd Edition). Prentice-Hall, 2008.
2. Charniak, E.: Statistical Language Learning. The MIT Press. 1996.
3. J. Allen: Natural Language Understanding. Benjamin/Cummins.1995.

Computational Linguistics Lab

Practical: 60 Lectures

Practical exercises based on concepts listed in theory.

V. Digital Image Processing

Theory: 60 Lectures

1. Introduction (6 Lectures)

Light, Brightness adaption and discrimination, Pixels, coordinate conventions, Imaging Geometry, Perspective Projection, Spatial Domain Filtering, sampling and quantization.

2. Spatial Domain Filtering (7 Lectures)

Intensity transformations, contrast stretching, histogram equalization, Correlation and convolution, Smoothing filters, sharpening filters, gradient and Laplacian.

3. Filtering in the Frequency domain (8 Lectures)

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

4. Image Restoration (8 Lectures)

Basic Framework, Interactive Restoration, Image deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Adaptive filters, Linear, Position invariant degradations, Estimation of Degradation functions, Restoration from projections.

5. Image Compression (10 Lectures)

Encoder-Decoder model, Types of redundancies, Lossy and Lossless compression, Entropy of an information source, Shannon's 1st Theorem, Huffman Coding, Arithmetic Coding, Golomb Coding, LZW coding, Transform Coding, Sub-image size selection, blocking artifacts, DCT implementation using FFT, Run length coding, FAX compression (CCITT Group-3 andGroup-4), Symbol-based coding, JBIG-2, Bit-plane encoding, Bit-allocation, Zonal Coding, Threshold Coding, JPEG, Lossless predictive coding, Lossy predictive coding, Motion Compensation

6. Wavelet based Image Compression (5 Lectures)

Expansion of functions, Multi-resolution analysis, Scaling functions, MRA refinement equation, Wavelet series expansion, Discrete Wavelet Transform (DWT), Continuous Wavelet Transform, Fast Wavelet Transform, 2-D wavelet Transform, JPEG-2000 encoding, Digital Image Watermarking.

7. Morphological Image Processing (7 Lectures)

Basics, SE, Erosion, Dilation, Opening, Closing, Hit-or-Miss Transform, Boundary Detection, Hole filling, Connected components, convex hull, thinning, thickening, skeletons, pruning, Geodesic Dilation, Erosion, Reconstruction by dilation and erosion.

8. Image Segmentation (9 Lectures)

Boundary detection based techniques, Point, line detection, Edge detection, Edge linking, local processing, regional processing, Hough transform, Thresholding, Iterative thresholding, Otsu's
method, Moving averages, Multivariable thresholding, Region-based segmentation, Watershed algorithm, Use of motion in segmentation

Reference Books

1. R C Gonzalez , R E Woods, Digital Image Processing, 3rd Edition, Pearson Education.2008.
2. A K Jain, Fundamentals of Digital image Processing, Prentice Hall of India.1989.
3. K R Castleman, Digital Image Processing, Pearson Education.1996
4. Schalkoff, Digital Image Processing and Computer Vision, John Wiley and Sons.1989.
5. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins,' Digital Image Processing using MATLAB', Pearson Education, Inc., 2004.

Digital Image Processing Lab
Practical: 60 Lectures

1. Write program to read and display digital image using MATLAB or SCILAB
    a. Become familiar with SCILAB/MATLAB Basic commands
    b. Read and display image in SCILAB/MATLAB

c. Resize given image
d. Convert given color image into gray-scale image
e. Convert given color/gray-scale image into black & white image
f. Draw image profile
g. Separate color image in three R G & B planes
h. Create color image using R, G and B three separate planes
i. Flow control and LOOP in SCILAB
j. Write given 2-D data in image file

2. To write and execute image processing programs using point processing method
   a. Obtain Negative image
   b. Obtain Flip image
   c. Thresholding
   d. Contrast stretching

3. To write and execute programs for image arithmetic operations
   a. Addition of two images
   b. Subtract one image from other image
   c. Calculate mean value of image
   d. Different Brightness by changing mean value

4. To write and execute programs for image logical operations
   a. AND operation between two images
   b. OR operation between two images
   c. Calculate intersection of two images
   d. Water Marking using EX-OR operation

e. NOT operation (Negative image)

5. To write a program for histogram calculation and equalization using
   a. Standard MATLAB function
   b. Program without using standard MATLAB functions
   c. C Program

6. To write and execute program for geometric transformation of image
   a. Translation
   b. Scaling
   c. Rotation
   d. Shrinking
   e. Zooming

7. To understand various image noise models and to write programs for
   a. image restoration
   b. Remove Salt and Pepper Noise
   c. Minimize Gaussian noise
   d. Median filter and Weiner filter

8. Write and execute programs to remove noise using spatial filters
   a. Understand 1-D and 2-D convolution process
   b. Use 3x3 Mask for low pass filter and high pass filter

9. Write and execute programs for image frequency domain filtering
   a. Apply FFT on given image
   b. Perform low pass and high pass filtering in frequency domain
   c. Apply IFFT to reconstruct image

10. Write a program in C and MATLAB/SCILAB for edge detection using different edge detection mask
11. Write and execute program for image morphological operations erosion and dilation.
12. To write and execute program for wavelet transform on given image and perform inverse wavelet transform to reconstruct image.

VI. Machine Learning

Theory: 60 Lectures

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, Backpropagation 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

Practical: 60 Lectures

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 specificvalues 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 backpropagation - algorithm to predict the value of a variable based on the dataset of problem 14.

VII. Introduction to Data Science

Theory: 60 Lectures

Data Scientist's Tool Box: Turning data into actionable knowledge, introduction to the tools that will be used in building data analysis software: version control, markdown, git, GitHub, R, and RStudio.

R Programming Basics: Overview of R, R data types and objects, reading and writing data, Control structures, functions, scoping rules, dates and times, Loop functions, debugging tools, Simulation, code profiling

Getting and Cleaning Data: Obtaining data from the web, from APIs, from databases and from colleagues in various formats. basics of data cleaning and making data ―tidy‖.

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

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

Reference Books

1. Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Frontiline" by Schroff/O'Reilly, 2013.

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

3. John W. Foreman, "Data Smart: Using data Science to Transform Information into Insight" by John Wiley & Sons, 2013.

4. Ian Ayres, "Super Crunchers: Why Thinking-by-Numbers Is the New Way to Be Smart" Ist Edition by Bantam, 2007.

5. Eric Seigel, "Predictive Analytics: The Power to Predict who Will Click, Buy, Lie, or Die", 1st Edition, by Wiley, 2013.

6. Matthew A. Russel, "Mining the Social Web: Data mining Facebook, Twitter, Linkedln, Goole+, GitHub, and More", Second Edition, by O'Reilly Media, 2013.

Introduction to Data Science Lab

Practical: 60 Lectures

1. Write a program that prints _Hello World_ to the screen.
2. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n
3. Write a program that prints a multiplication table for numbers up to 12.
4. Write a function that returns the largest element in a list.
5. Write a function that computes the running total of a list.
6. Write a function that tests whether a string is a palindrome.
7. Implement linear search.
8. Implement binary search.
9. Implement matrices addition , subtraction and Multiplication
10. Fifteen students were enrolled in a course. There ages were:

    20 20 20 20 20 21 21 21 22 22 22 22 23 23 23

    i. Find the median age of all students under 22 years ii. Find the median age of all students

    iii. Find the mean age of all students iv. Find the modal age for all students

    v. Two more students enter the class. The age of both students is 23. What is now mean, mode and median ?

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

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

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

VIII. Cloud Computing

Overview of Computing Paradigm (8 lectures)

Recent trends in Computing : Grid Computing, Cluster Computing, Distributed Computing,

Utility Computing, Cloud Computing,

Introduction to Cloud Computing (7 lectures)

Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing,

Cloud Computing Architecture (20 lectures) 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.

Case Studies **(13 lectures)**

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

Service Management in Cloud Computing (7 lectures)

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

Cloud Security (5 lectures)

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

Reference Books

1. *Cloud Computing Bible,* Barrie Sosinsky, *Wiley-India,* 2010
2. *Cloud Computing: Principles and Paradigms,* Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, *Wile,* 2011
3. *Cloud Computing: Principles, Systems and Applications, Editors:* Nikos Antonopoulos, Lee Gillam, *Springer,* 2012
4. *Cloud Security: A Comprehensive Guide to Secure Cloud Computing,* Ronald L. Krutz, Russell Dean Vines, *Wiley-India,* 2010
5. Gautam Shroff, *Enterprise Cloud Computing Technology Architecture Applications* , Adobe Reader ebooks available from eBooks.com,2010
6. Toby Velte, Anthony Velte, Robert Elsenpeter, *Cloud Computing, A Practical Approach* ,McGraw Hills, 2010.
7. Dimitris N. Chorafas, *Cloud Computing Strategies* ,CRC Press, 2010

Cloud Computing Lab

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

Numerical Methods Theory: 60 Lectures

Floating point representation and computer arithmetic, Significant digits, Errors: Round-off error, Local truncation error, Global truncation error, Order of a method, Convergence and terminal conditions, Efficient computations

Bisection method, Secant method, Regula−Falsi method
Newton−Raphson method, Newton_s method for solving nonlinear systems
Gauss elimination method (with row pivoting) and Gauss−Jordan method, Gauss Thomas method for tridiagonal systems
Iterative methods: Jacobi and Gauss-Seidel iterative methods
Interpolation: Lagrange_s form and Newton_s form
Finite difference operators, Gregory Newton forward and backward differences Interpolation
Piecewise polynomial interpolation: Linear interpolation, Cubic spline interpolation (only method), Numerical differentiation: First derivatives and second order derivatives, Richardson extrapolation
Numerical integration: Trapezoid rule, Simpson_s rule (only method), Newton−Cotes open formulas
Extrapolation methods: Romberg integration, Gaussian quadrature, Ordinary differential equation: Euler_s method

Modified Euler‗s methods: Heun method and Mid-point method, Runge-Kutta second methods: Heun method without iteration, Mid-point method and Ralston‗s method

Classical 4th order Runge-Kutta method, Finite difference method for linear ODE

REFERNCE BOOKS:

[1] Laurence V. Fausett, Applied Numerical Analysis, Using MATLAB, Pearson, 2/e (2012)

[2] M.K. Jain, S.R.K. Iyengar and R.K. Jain, Numerical Methods for Scientific and Engineering Computation, New Age International Publisher, 6/e (2012)

[3] Steven C Chapra, Applied Numerical Methods with MATLAB for Engineers and Scientists, Tata McGraw Hill, 2/e (2010)

Numerical Methods Lab
Practical: 60 lectures

1. Find the roots of the equation by bisection method.
2. Find the roots of the equation by secant/Regula−Falsi method.
3. Find the roots of the equation by Newton‗s method.
4. Find the solution of a system of nonlinear equation using Newton‗s method.
5. Find the solution of tridiagonal system using Gauss Thomas method.
6. Find the solution of system of equations using Jacobi/Gauss-Seidel method.
7. Find the cubic spline interpolating function.
8. Evaluate the approximate value of finite integrals using Gaussian/Romberg integration.
9. Solve the boundary value problem using finite difference method.

Note: Programming is to be done in any one of Computer Algebra Systems:
MATLAB / MATHEMATICA / MAPLE.

X. Systems Programming

Theory: 60 lectures

Assemblers & Loaders, Linkers: 10L

One pass and two pass assembler, design of an assembler, Absolute loader, relocation and linking concepts, relocating loader and Dynamic Linking.

Introduction: 2L

Overview of compilation, Phases of a compiler

Lexical Analysis: 6L

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

Parsing: 10L

Bottom up parsing- LR parser, yacc.

Intermediate representations 10L

Three address code generation, syntax directed translation, translation of types, control statements

Storage organization: 5L

Activation records, stack allocation

Code Generation: 5L

Object code generation

Reference Books

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

SYSTEMS PROGRAMMING LAB
Practical: 60 lectures

[1] To implement an assembler for a hypothetical language.

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

[3] To get familiar with yacc: write a desk calculator.

XI. Operational Research

Theory: 60 Lectures

Introduction to Operational Research (OR): Origin & Development, Different Phases of OR study, Methodology of OR, Scope and Limitations of OR, Applications of OR.

Linear Programming: Linearly independent / dependent vectors, Basis, Convex sets, Extreme points. Graphical method. Simplex method, Artificial variable techniques- Two

Phase Method; M-Charnes Method, Special cases in LPP.

Duality: Definition of the dual problem, Primal-dual relationships, Economic Interpretation of Duality, Dual simplex Method.

Sensitivity analysis: Changes in cost and resource vector

Reference Books

1. **G. Hadley: Linear Programming. Narosa, 2002 (reprint).**

2. **A. Ravindran, D. T. Phillips and James J. Solberg:** Operations Research-Principles and Practice, John Wiley & Sons, 2005.

3. **Hamdy A. Taha:** Operations Research-An Introduction, Prentice Hall, 8th Edition, 2008.

4. **F.S. Hillier. G.J. Lieberman**: Introduction to Operations Research- Concepts and Cases, 9th Edition, Tata McGraw Hill. 2010.

Operational Research Lab
Practical: 60 Lectures

[1] To solve Linear Programming Problem using Graphical
Method with Unbounded solution
Infeasible solution
Alternative or multiple solutions.

[2] Solution of LPP with simplex method.

[3] Problem solving using M-Charnes method.

[4] Problem solving using Two Phase method.

[5] Illustration of following special cases in LPP using Simplex method

Unrestricted variables

Unbounded solution

Infeasible solution

Alternative or multiple solution

[6] Problems based on Dual simplex method.

[7] Problems based on sensitivity analysis.

XII. Data Mining

Theory: 60 lectures

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

Data Mining Techniques: Association Rule Mining, classification and regression techniques, clustering, Scalability and data management issues in data mining algorithms, measures of interestingness

Books Recommended:

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

Data Mining Lab

Practical: 60 lectures

Practical exercises based on concepts listed in theory.

XIII. Discipline Specific Elective: Dissertation / Project work

This option to be offered only in 6th Semester.
The students will be allowed to work on any project based on the concepts studied in core / elective or skill based elective courses.

The group size should be maximum of three (03) students.

Each group will be assigned a teacher as a supervisor who will handle both their theory as well lab classes.

A maximum of Four (04) projects would be assigned to one teacher.

Skill Enhancement Courses (any two) (Credit: 02 each) – SEC1 to SEC2 Theory: 01, Labs: 02

1. Android Programming

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

(2L)

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

Development Tools: Installing and using Eclipse with ADT plug-in, Installing Virtual machine for Android sandwich/Jelly bean (Emulator), configuring the installed tools, creatinga android project

– Hello Word, run on emulator, Deploy it on USB-connected Android device.

(5L)

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

(2L)

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

(2L)

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

Book Recommended:

1. Android application development for java programmers. By James C. Sheusi. Publisher: Cengage Learning, 2013.

ONLINE READING / SUPPORTING MATERIAL:

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

Software Lab Based on Android Programming:

1. Create ―Hello World‖ application. That will display ―Hello World‖ in the middle of the screen in the emulator. Also display ―Hello World‖ in the middle of the screen in the Android Phone.
2. Create an application with login module. (Check username and password).
3. Create spinner with strings taken from resource folder (res >> value folder) and on changing the spinner value, Image will change.
4. Create a menu with 5 options and and selected option should appear in text box.
5. Create a list of all courses in your college and on selecting a particular course teacher-

in-charge of that course should appear at the bottom of the screen.

6. Create an application with three option buttons, on selecting a button colour of the screen will change.
7. Create and Login application as above. On successful login, pop up the message.
8. Create an application to Create, Insert, update, Delete and retrieve operation on the database.

2. Programming in MATLAB (1+2Labs)

Unit I- Introduction to Programming: Components of a computer, working with numbers, Machine code, Software hierarchy. (2L)

Unit I- 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 matrixI=[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 + 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

•

HTML Programming

- **Unit-I: Introduction** (1L)
- **Unit-II: The Basics** (2L)
  - The Head, the Body
  - Colors, Attributes
  - Lists, ordered and unordered
- **Unit-III: Links** (3L)
  - Introduction
  - Relative Links, Absolute Links
  - Link Attributes
  - Using the ID Attribute to Link Within a Document
- **Unit-IV: Images** (2L)
  - Putting an Image on a Page
  - Using Images as Links
  - Putting an Image in the Background
- **Unit V: – Tables** (4L)
  - Creating a Table
  - Table Headers
  - Captions
  - Spanning Multiple Columns
  - Styling Table
- **Unit VI – Forms** (3L)
  - Basic Input and Attributes
  - Other Kinds of Inputs ○ Styling forms with CSS
  - Where To Go From Here

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

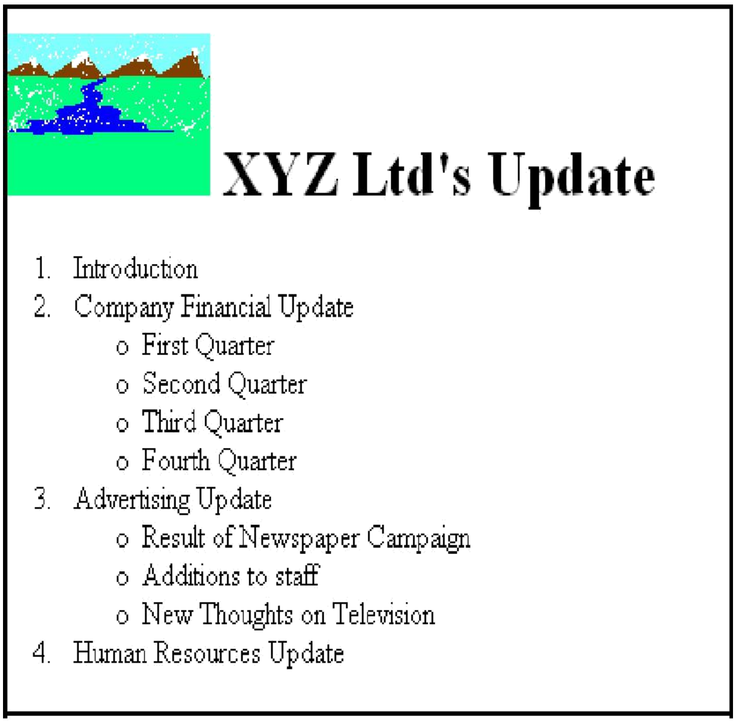
Software Lab Based on HTML:

Create an HTML document with the following formatting options:

I. Bold
II. Italics
III. Underline
IV. Headings (Using H1 to H6 heading styles)
V. Font (Type, Size and Color)
VI. Background (Colored background/Image in background)
VII. Paragraph
VIII. Line Break
IX. Horizontal Rule
X. Pre tag

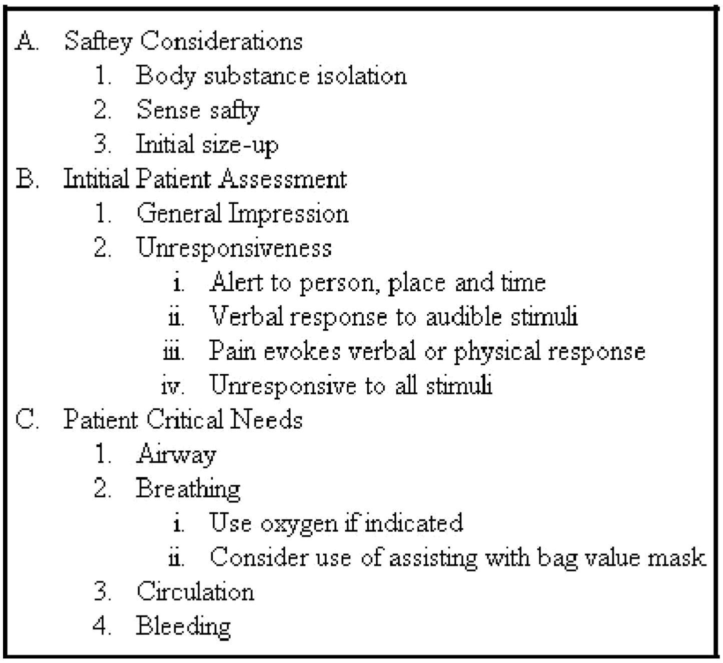
Create an HTML document which consists of:

I. Ordered List
II. Unordered List III. Nested List
IV. Image



XYZ Ltd's Update

1. Introduction
2. Company Financial Update
   - o First Quarter
   - o Second Quarter
   - o Third Quarter
   - o Fourth Quarter
3. Advertising Update
   - o Result of Newspaper Campaign
   - o Additions to staff
   - o New Thoughts on Television
4. Human Resources Update



A. Saftey Considerations
   1. Body substance isolation
   2. Sense safty
   3. Initial size-up

B. Intitial Patient Assessment
   1. General Impression
   2. Unresponsiveness
      i. Alert to person, place and time
      ii. Verbal response to audible stimuli
      iii. Pain evokes verbal or physical response
      iv. Unresponsive to all stimuli

C. Patient Critical Needs
   1. Airway
   2. Breathing
      i. Use oxygen if indicated
      ii. Consider use of assisting with bag value mask
   3. Circulation
   4. Bleeding

Create an HTML document which implements Internal linking as well as External linking. Q.4
Q4 Create a table using HTML which consists of columns for Roll No., Student_s name and grade.

Result		
Roll No.	Name	Grade

Create a Table with the following view:

Q.6

<table>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="3"></td><td colspan="2" rowspan="3">Place an image here</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3"></td></tr>
</table>

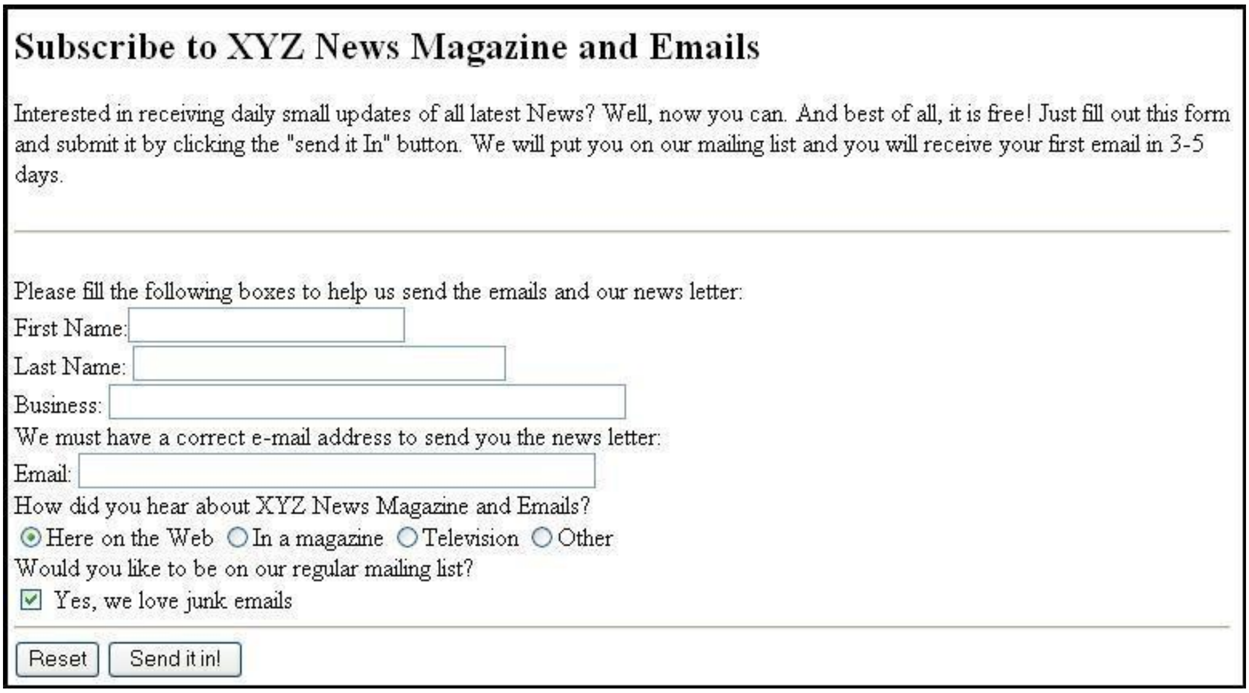
Create a form using HTML which has the following types of controls:

I. Text Box

II. Option/radio buttons

III. Check boxes

IV. Reset and Submit buttons



Subscribe to XYZ News Magazine and Emails

Interested in receiving daily small updates of all latest News? Well, now you can. And best of all, it is free! Just fill out this form and submit it by clicking the "send it In" button. We will put you on our mailing list and you will receive your first email in 3-5 days.

Please fill the following boxes to help us send the emails and our news letter:

First Name:

Last Name:

Business:

We must have a correct e-mail address to send you the news letter:

Email:

How did you hear about XYZ News Magazine and Emails?

● Here on the Web ○ In a magazine ○ Television ○ Other

Would you like to be on our regular mailing list?

☑ Yes, we love junk emails

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

Frame1
Frame2

<table>
<tr><td colspan="2">Frame1</td></tr>
<tr><td>Frame2</td><td>Frame3</td></tr>
</table>

4. XML Programming

Introduction: Understanding Mark-up Languages, Introduction to XML and its Goals. (3L)

XML Basics: XML Structure and Syntax, Document classes and Rules. (5L)

Other XML Concepts: Scripting XML, XML as Data, Linking with XML. (4L)

XML with Style: XSL –Style Sheet Basics, XSL basics, XSL style sheets. (3L)

Books Recommended

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

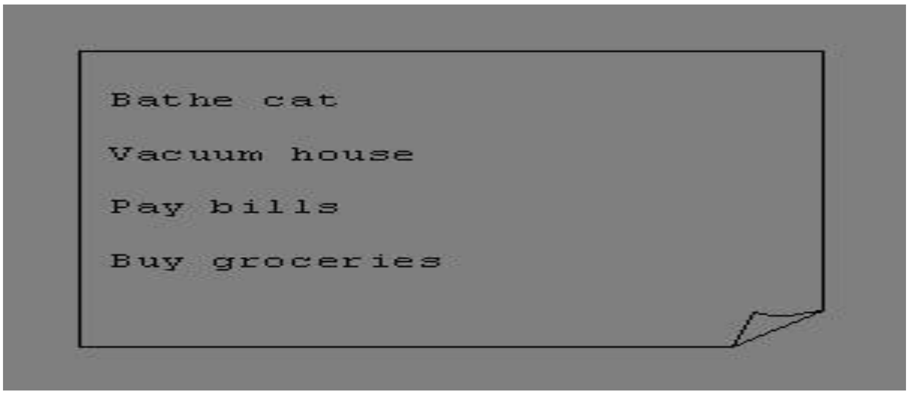
Software Lab Based on XML:

Exercise #1 – Information Structure

In this exercise, student will practice identifying the structure of an information object. For the sample document provided below:

Label the information structures you see, including containing structures.

1. Draw a tree representation of the structure.



Bathe cat

Vacuum house

Pay bills

Buy groceries

Exercise 2# Deconstructing an XML Document

In this exercise, student will practice identifying the explicit structure within an XML document. In a sense, this is the reverse of what you did in Exercise #1. For the sample XML markup below, create a document-like representation (or a simple drawing) for the content contained within the XML tags:

```
<book>

<coverInfo>

    <title>The XML Handbook</title>

    <author>Charles F. Goldfarb</author>

    <author>Paul Prescod</author>

    <edition>Second</edition>
```

<description>The definitive XML resource: applications, products, and technologies. Revised and expanded—over 600 new pages.
</description>

</coverInfo>

</book>

Exercise #3 – Creating XML Markup

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

Exercise #4 – Well-Formedness

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

<list><title>The first list</title><item>An item</list>

<item>An item</item><item>Another item</item>

<para>Bathing a cat is a <emph>relatively</emph> easy task as long as the cat is willing.</para>

<bibl><title>How to Bathe a Cat<author></title>Merlin Bauer<author></bibl>

Exercise #5-Well Formedness

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

<PROCEDURE><TITLEHow to Bathe a Cat</TITLE>

<OVERVIEW>

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

Note: Cover more exercises based on XML Programming theory concepts.

5. Oracle (SQL/PL-SQL) (1+2 Lab)

Introduction to Oracle as RDBMS

SQL Vs. SQL * Plus:

SQL Commands and Data types, Operators and Expressions, Introduction to SQL * Plus. (2L)
Managing Tables and Data:

- Creating and Altering Tables (Including constraints)
- Data Manipulation Command like Insert, update, delete
- SELECT statement with WHERE, GROUP BY and HAVING, ORDER BY, DISTINCT, Special operator like IN, ANY, ALL BETWEEN, EXISTS, LIKE
- Join, Built in functions (4L)

Other Database Objects

- View
- Synonyms, Index (2L)

Transaction Control Statements

- Commit, Rollback, Savepoint (2L)

Introduction to PL/SQL

- SQL v/s PL/SQL
- PL/SQL Block Structure
- Language construct of PL/SQL (Variables, Basic and Composite Data type, Conditions looping etc.)
- % TYPE and % ROWTYPE
- Using Cursor (Implicit, Explicit) (5L)

Books Recommended:

1. Ivan Bayross, "SQL, PL/SQL the Programming Language of Oracle Paperback", BPB Publicatins, 2010.

2. Steven Feuerstein, Bill Pribyl , "Oracle PL/SQL Programming", 6th Edition, O'Reilly Media, 2014.

3. Rajeeb C. Chatterjee, "Learning Oracle SQL and PL/SQL: A simplified Guide", PHI, 2012.

4. Ron Hardman, Michael Mclaughlin, "Expert Oracle PL/SQL", Oracle Press, 2005.

5. Michael Mclaughlin, "Oracle Database 11g PL/SQL Programming", Oracle Press, 2008.

6. John Watson, Roopesh Ramklass, "OCA Oracle Database11g SQL Fundamentals I Exam Guide", Oracle Press, 2008.

Software Lab Based on SQL/PL-SQL:

[SQL COMMANDS]

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

[SQL FUNCTION]

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

[PL/SQL]

11) To understand working with PL/SQL

12) To implement Cursor on a table.
13) To implement trigger on a table

Increment or Decrement operator). (4L)

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

(4L)

Reference Books

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. Python Tutorial/Documentation www.python.or 2015
3. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python , Freely available online.2012
4. http://docs.python.org/3/tutorial/index.html
5. http://interactivepython.org/courselib/static/pythonds
6. http://www.ibiblio.org/g2swap/byteofpython/read/

Software Lab Based on Python:

Section: A (Simple programs)

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users choice.
2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria :

   Grade A: Percentage >=80

   Grade B: Percentage>=70 and
   <80 Grade C: Percentage>=60
   and <70 Grade D:
   Percentage>=40 and <60 Grade
   E: Percentage<40
3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input paramters from user.
4. WAP to display the first n terms of Fibonacci series.
5. WAP to find factorial of the given number.
6. WAP to find sum of the following series for n terms: 1 – 2/2! + 3/3! -------- n/n!
7. WAP to calculate the sum and product of two compatible matrices.

Section: B (Visual Python):

All the programs should be written using user defined functions, wherever possible.

1. Write a menu-driven program to create mathematical 3D objects I. curve
   - II. sphere
   - III. cone
   - IV. arrow
   - V. ring
   - VI. cylinder.
2. WAP to read n integers and display them as a histogram.
3. WAP to display sine, cosine, polynomial and exponential curves.
4. WAP to plot a graph of people with pulse rate p vs. height h. The values of p and h are to be entered by the user.
5. WAP to calculate the mass m in a chemical reaction. The mass m (in gms) disintegrates according to the formula m=60/(t+2), where t is the time in hours. Sketch a graph for t vs. m, where t>=0.
6. A population of 1000 bacteria is introduced into a nutrient medium. The population p grows as follows:

   P(t) = (15000(1+t))/(15+ *e)*

   where the time t is measured in hours. WAP to determine the size of the population at given time t and plot a graph for P vs t for the specified time interval.
7. Input initial velocity and acceleration, and plot the following graphs depicting equations of motion:
   - I. velocity wrt time (v=u+at)
   - II. distance wrt time (s=u*t+0.5*a*t*t)
   - III. distance wrt velocity (s=(v*v-u*u)/2*a)
8. WAP to show a ball bouncing between 2 walls. (Optional)

7. PHP Programming (1 +2 Lab)

Introduction to PHP: (3L)

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

PHP with other technologies, scope of PHP

Basic Syntax, PHP variables and constants

Types of data in PHP , Expressions, scopes of a variable (local, global)

PHP Operators : Arithmetic, Assignment, Relational , Logical operators, Bitwise , ternary and MOD operator.

PHP operator Precedence and associativity

Handling HTML form with PHP (2L)

Capturing Form Data

GET and POST form methods

Dealing with multi value fields

Redirecting a form after submission

PHPconditional events and Loops **(3L)**

PHP IF Else conditional statements (Nested IF and Else)
Switch case, while ,For and Do While Loop
Goto , Break ,Continue and exit

PHP Functions: **(3L)**

Function, Need of Function , declaration and calling of a function

PHP Function with arguments, Default Arguments in Function
Function argument with call by value, call by reference
Scope of Function Global and Local

String Manipulation and Regular Expression: **(3L)**

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

Array: **(3L)**

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

Reference Books:

1. Steven Holzner, "PHP: The Complete Reference Paperback", McGraw Hill Education (India), 2007.

2. Timothy Boronczyk, Martin E. Psinas, "PHP and MYSQL (Create-Modify-Reuse)", Wiley India Private Limited, 2008.

3. Robin Nixon, "Learning PHP, MySQL, JavaScript, CSS & HTML5", 3rd Edition Paperback, O'reilly, 2014.

4. Luke Welling, Laura Thompson, PHP and MySQL Web Development", 4th Edition, Addition Paperback, Addison-Wesley Professsional,2008.

5. David Sklar, Adam Trachtenberg, "PHP Cookbook: Solutions & Examples for PHP Programmers", 2014.

Software Lab Based on PHP:

1. Create a PHP page using functions for comparing three integers and print the largest number.
2. Write a function to calculate the factorial of a number (non-negative integer). The function accept the number as an argument.
3. WAP to check whether the given number is prime or not.
4. Create a PHP page which accepts string from user. After submission that page displays the reverse of provided string.

5. Write a PHP function that checks if a string is all lower case.
6. Write a PHP script that checks whether a passed string is palindrome or not? (A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run)
7. WAP to sort an array.
8. Write a PHP script that removes the whitespaces from a string.

   Sample string : 'The quick " " brown fox' Expected Output : Thequick""brownfox
9. Write a PHP script that finds out the sum of first n odd numbers.
10. Create a login page having user name and password. On clicking submit, a welcome message should be displayed if the user is already registered (i.e.name is present in the database) otherwise error message should be displayed.
11. Write a PHP script that checks if a string contains another string.
12. Create a simple 'birthday countdown' script, the script will count the number of days between current day and birth day.
13. Create a script to construct the following pattern, using nested for loop.

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

14. Write a simple PHP program to check that emails are valid.
15. WAP to print first n even numbers.
16. $color = array('white', 'green', 'red")

    Write a PHP script which will display the colors in the following way : Output :

    white, green, red,

    - green
    - red
    - white
17. Using switch case and dropdown list display a ―Hello‖ message depending on the language selected in drop down list.
18. Write a PHP program to print Fibonacci series using recursion.
19. Write a PHP script to replace the first 'the' of the following string with 'That'.

    Sample : 'the quick brown fox jumps over the lazy dog.'

    Expected Result : That quick brown fox jumps over the lazy dog.

8. Linux / Unix Programming Syllabus (1+2 Labs)

Introduction (5 L)

What is linux/unix Operating systems
Difference between linux/unix and other operating systems
Features and Architecture
Various Distributions available in the market
Installation, Booting and shutdown process

ystem processes (an overview)
External and internal commands

Creation of partitions in OS
Processes and its creation phases – Fork, Exec, wait

User Management and the File System

Types of Users, Creating users, Granting rights
User management commands
File quota and various file systems available
File System Management and Layout, File permissions
Login process, Managing Disk Quotas
Links (hard links, symbolic links)

introduction Scripting and Shell (6L)

What is shell and various type of shell, Various editors present in linux

Different modes of operation in vi editor

What is shell script, Writing and executing the shell script

Shell variable (user defined and system variables)

System calls, Using system calls

Pipes and Filters

Decision making in Shell Scripts (If else, switch), Loops in shell

Functions

Utility programs (cut, paste, join, tr , uniq utilities)

Pattern matching utility (grep)

Reference Books:

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

sockets Networking API, Vol. 1, 3rd Edition,2014

Software Lab Based on Linux:

1. Write a shell script to check if the number entered at the command line is prime or not.
2. Write a shell script to modify ―cal‖ command to display calendars of the specified months.
3. Write a shell script to modify ―cal‖ command to display calendars of the specified range of months.
4. Write a shell script to accept a login name. If not a valid login name display message – ―Entered login name is invalid‖.
5. Write a shell script to display date in the mm/dd/yy format.
6. Write a shell script to display on the screen sorted output of ―who‖ command along with the total number of users .

7. Write a shell script to display the multiplication table any number,
8. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
9. Write a shell script to find the sum of digits of a given number.

10. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
11. Write a shell script to find the LCD(least common divisor) of two numbers.
12. Write a shell script to perform the tasks of basic calculator.
13. Write a shell script to find the power of a given number.
14. Write a shell script to find the binomial coefficient C(n , x).
15. Write a shell script to find the permutation P(n,x).
16. Write a shell script to find the greatest number among the three numbers.
17. Write a shell script to find the factorial of a given number.
18. Write a shell script to check whether the number is Armstrong or not.
19. Write a shell script to check whether the file have all the permissions or not.

Program to show the pyramid of special character —*‖.**9.**

R-Programming (1+2 Labs)

Introduction: Overview and History of R, Getting Help, Data Ty pes, Subsetting, Vectorized Operations, Reading and Writing Data. **(5L)**

Control Structures, Functions, lapply, tapply, split, mapply, apply, Coding Standards. **(5L)**

Scoping Rules, Debugging Tools, Simulation, R Profiler. **(5L)**

Reference Books

1. William N. Venables and David M. Smith, An Introduction to R. 2nd Edition. Network Theory Limited.2009

2. Norman Matloff, The Art of R Programming - A Tour of Statistical Software Design, No Starch Press.2011

Software Lab Based on R Programming

1. Write a program that prints ‗Hello World‗ to the screen.
2. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n
3. Write a program that prints a multiplication table for numbers up to 12.
4. Write a function that returns the largest element in a list.
5. Write a function that computes the running total of a list.
6. Write a function that tests whether a string is a palindrome.
7. Implement the following sorting algorithms: Selection sort, Insertion sort, Bubble Sort
8. Implement linear search.
9. Implement binary search.
10. Implement matrices addition, subtraction and Multiplication

10. Software Testing (1+2 Labs)

Introduction **(4L)**

Strategic Approach to Software Testing, Test Strategies for Conventional Software, Validation Testing, System Testing, Basic Terminologies, V Shaped Software Lifecycle Model

Functional Testing\ Black-box Testing **(5L)**

Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing

Structural Testing\ White-box Testing **(6L)**

Basis Path Testing: Program Graph, DD Path graph, Cyclomatic Complexity, Graph Matrices, Control Flow Testing: Statement Coverage, Branch Coverage, Condition Coverage, Path Coverage

Books Recommended:

1. Roger S. Pressman, Software Engineering: A Practitioner_s Approach, Seventh Edition, Mc Graw Hill Education.2009

2. Yogesh Singh, Software Testing, Cambridge University Press.2011.

Computer Lab Based on Software Testing:

1. Write a program that take three inputs (a,b &c) that represent the sides of a triangle, and the output is one of the below four:

   a. Not a triangle

   b. Scalene triangle

   c. Isosceles triangle

   d. Equilateral triangle

   Generate test cases using Boundary Value Analysis, Equivalence Class Partitioning and Decision Table Testing.

   Generate test cases using Basis path testing.

   Run code coverage tool.

2. Write a program that determines the nature of roots of a quadratic equation. Output should be one of the following:-

   - Not a quadratic equation.
   - Complex roots
   - Real roots
   - Single roots

   I. Generate test cases using Boundary Value Analysis, Equivalence Class Partitioning and Decision Table Testing.

   II. Generate test cases using Basis path testing.

   III. Run code coverage tool

3. Write a program that checks whether the number is even or odd. Run code coverage tool and find the amount of code being covered.

4. Write a program that dynamically allocates memory to10 integers using malloc() or calloc() and

   - donot free memory leading to memory leaks. Verify the same usingValgrind.
   - Now, free memory using free() at the end of the program to avoid memory leaks. Verify the same using Valgrind.

5. Use LoadUI load testing tool to test the web application performance.

Syllabus

For

General Elective (GE) (Minor– Computer Science)

(Choice Based Credit System)

General Elective Papers (GE) (Minor – Computer Science)

GE1:

Computer Fundamentals

Computer Fundamentals LAB

GE2:

Introduction to Programming

Introduction to Programming LAB

GE3

Computer Fundamentals and Python Programming

Fundamental and Programming in Python Lab

GE4:

Data Structure using Python

Data Structure LAB

Computer Fundamentals
Theory: 60 lectures

Introduction: Introduction to computer system, uses, types. **6L**

Data Representation: Number systems and character representation, binary **12L**

arithmetic

Human Computer Interface: Types of software, Operating system as user interface, utility programs **6L**

Devices: Input and output devices (with connections and practical demo), keyboard, mouse, joystick, scanner, OCR, OMR, bar code reader, web camera, monitor, printer, plotter **6L**

Memory: Primary, secondary, auxiliary memory, RAM, ROM, cache memory, hard disks, optical disks

Computer Organisation and Architecture: C.P.U., registers, system bus, main memory unit, cache memory, Inside a computer, SMPS, Motherboard, Ports and Interfaces, expansion cards, ribbon cables, memory chips, processors. **12L**

Overview of Emerging Technologies: Bluetooth, cloud computing, big data, data mining, mobile computing and embedded systems. **8L**

Reference Books:

1. A. Goel, Computer Fundamentals, Pearson Education, 2010.
2. P. Aksoy, L. DeNardis, Introduction to Information Technology, Cengage Learning, 2006
3. P. K.Sinha, P. Sinha, Fundamentals of Computers, BPB Publishers, 2007

Computer Fundamentals Lab
Practical: 60 lectures

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

MS Word

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

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

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

3. Design a **time-table form** for your college.
   - The first line should mention the name of the college in 16-point Arial Font and should be bold.
   - □ The second line should give the course name/teacher_s name and the department in 14-point Arial
   - Leave a gap of 12-points.
   - The rest of the document should use 10-point Times New Roman font.
   - The footer should contain your specifications as the designer and date of creation.

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

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

5. Create the following one page documents.

a. Compose a note inviting friends to a get-together at your house, Including a list of things to bring with them.

b. Design a certificate in landscape orientation with a border around the document.

c. Design a Garage Sale sign.

d. Make a sign outlining your rules for your bedroom at home, using a numbered list.

6. Create the following documents:

(a) A newsletter with a headline and 2 columns in portrait orientation, including at least one image surrounded by text.

(b) Use a newsletter format to promote upcoming projects or events in your classroom or college.

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

Type the following as shown (do not bold).

Color, Style, Item
Blue, A980, Van
Red, X023, Car
Green, YL724, Truck
Name, Age, Sex

Bob, 23, M

Linda, 46, F

Tom, 29, M

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

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

Banks, Jennifer	1201	2528	1203
Atwater, Kelly	4098	3079	2067

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

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

10. Wrapping of text around the image.

11.Following features of menu option must be coveredFILE

	Complete menu
EDIT	Complete menu
VIEW	Complete menu
INSERT	Complete menu
FORMAT	Complete menu
TABLE	Complete menu
WINDOW	Complete menu
HELP	Complete menu
TOOLS	All options except Online collaboration, Tools on Macro, Templates

MS Excel

1. Enter the Following data in Excel Sheet

REGIONAL SALES PROJECTION

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

(a) Apply Formatting as follow:

I.Title in TIMES NEW ROMAN

ii. Font Size - 14

iii. Remaining text - ARIAL, Font Size -10

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

v. Numbers in two decimal places.

vi. Qtr. Heading in center Alignment.

vii. Apply Border to whole data.

(b) Calculate State and Qtr. Total

(c) Calculate Average for each quarter

(d) Calculate Amount = Rate * Total.

2. Given the following worksheet

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

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

If Marks	Then Grade
>= 80	A+
>= 60 < 80	A
>= 50 < 60	B
< 50	F

3. Given the following worksheet

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

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

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

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

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

Allowances

- HRA Dependent on Basic

30% of Basic if Basic <=1000

25% of Basic if Basic>1000 & Basic<=3000

20% of Basic if Basic >3000

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

Deductions

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

Calculate the following:

Gross Salary = Basic + HRA + DA + Conveyance + Entertainment

Total deduction = Provident Fund + Group Insurance Premium

Net Salary = Gross Salary – Total Deduction

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

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

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

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

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

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

(*a*) Calculate total sale year wise.

(*b*) Calculate the net sale made by each salesman

(*c*) Calculate the maximum sale made by the salesman

(d) Calculate the commission for each salesman under the condition.

(*i*) If total sales >4,00,000 give 5% commission on total sale made by the salesman.

(*ii*) Otherwise give 2% commission.

(*e*) Draw a bar graph representing the sale made by each salesman.

(*f*) Draw a pie graph representing the sale made by salesman in 2000.

8. Enter the following data in Excel Sheet

PERSONAL BUDGET FOR FIRST QUARTER

Monthly Income (Net): 1,475

EXPENSES	**JAN**	**FEB**	**MARCH**	**QUARTER TOTAL**	**QUARTER AVERAGE**
Rent600.00	600.00	600.00			
Telephone	48.25	43.50	60.00		
Utilities	67.27	110.00	70.00		
Credit Card	200.00	110.00	70.00		
Oil	100.00	150.00	90.00		
AV to Insurance	150.00				
Cable TV	40.75	40.75	40.75		
Monthly Total					

Calculate Quarter total and Quarter average.

(a) Calculate Monthly total.

(b) Surplus = Monthly income - Monthly total.

(c) What would be total surplus if monthly income is 1500.

(d) How much does telephone expense for March differ from quarter average.

(e) Create a 3D column graph for telephone and utilities.

(f) Create a pie chart for monthly expenses.

9. Enter the following data in Excel Sheet

TOTAL REVENUE EARNED FOR SAM'S BOOKSTALL

Publisher name	**1997**	**1998**	**1999**	**2000**	**total**
A	Rs. 1,000.00	Rs. 1100.00	Rs. 1,300.00	Rs. 800.00	
B	Rs. 1,500.00	Rs. 700.00	Rs. 1,000.00	Rs. 2,000.00	
C	Rs. 700.00	Rs. 900.00	Rs. 1,500.00	Rs. 600.00	
D	Rs. 1,200.00	Rs. 500.00	Rs. 200.00	Rs. 1,100.00	
E	Rs 800.00	Rs. 1,000.00	Rs. 3,000.00	Rs. 560.00	

(a) Compute the total revenue earned.

(b) Plot the line chart to compare the revenue of all publisher for 4 years.

(b) Chart Title should be _Total Revenue of sam_s Bookstall (1997-2000)_

(c) Give appropriate categories and value axis title.

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

Introduction to Programming

Theory: 60 lectures

Introduction to C and C++ (5 Lectures)

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

Data Types, Variables, Constants, Operators and Basic I/O (10 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, putcharetc), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.hetc).

Expressions, Conditional Statements and Iterative Statements (10 Lectures)

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

Functions and Arrays (10 Lectures)

Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments.

Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays

Derived Data Types (Structures and Unions) (5 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 data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.

File I/O, Preprocessor Directives (8 Lectures)

Opening and closing a file (use of fstream header file, ifstream, ofstream and fstream classes), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros

Using Classes in C++ (8 Lectures)

Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Function overloading in classes, Class Variables &Functions, Objects as parameters, Specifying the Protected and Private Access, Copy Constructors, Overview of Template classes and their use.

Inheritance and Polymorphism (4 Lectures)

Introduction to Inheritance and Polymorphism

1. HerbtzSchildt, "C++: The Complete Reference", Fourth Edition, McGraw Hill.2003
2. BjarneStroustrup, "The C++ Programming Language", 4th Edition, Addison-Wesley , 2013.
3. Addison-Wesley 2014.
4. E Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw-Hill Education, 2008.
5. Paul Deitel, Harvey Deitel, "C++ How to Program", 8th Edition, Prentice Hall, 2011.
6. John R. Hubbard, "Programming with C++", Schaum's Series, 2nd Edition, 2000.
7. Andrew Koeni, Barbara, E. Moo, "Accelerated C++", Published by Addison-Wesley , First Create space Inc, O-D Publishing, LLC USA.2014
10. Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.
11. Stanley B. Lippman, JoseeLajoie, Barbara E. Moo, "C++ Primer", Published by Addison-

Introduction to c/c++ Programming Lab

Practical: 60 lectures

1. Write a program to find greatest of three numbers.
2. Write a program to find gross salary of a person
3. Write a program to find grade of a student given his marks.
4. Write a program to find divisor or factorial of a given number.
5. Write a program to print first ten natural numbers.
6. Write a program to print first ten even and odd numbers.
7. Write a program to find grade of a list of students given their marks.
8. Create Matrix class. Write a menu-driven program to perform following Matrix operations (2-D array implementation):
   a) Sum b) Difference c) Product d) Transpose

Computer Fundamentals and Python Programming

Theory: 60 lectures

Introduction: Introduction to computer system, uses, types.

Data Representation: Number systems and character representation, binary arithmetic

Human Computer Interface: Types of software, Operating system as user interface, utility programs Machine language, Assembly language, High level language

Devices &Memory: Input and output devices Primary, secondary, auxiliary memory, RAM, ROM, cache memory, hard disks, optical disks

Computer Organisation and Architecture: C.P.U., registers, system bus, main memory unit, cache memory, BIOS, Utility Programs, File Maintenance, Language processors, Assembler, Compiler And Interpreter.

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

Creating Python Programs: Input and Output Statements, Control statements(Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments, Errors and Exceptions.

Iteration and Recursion: Conditional execution, Alternative execution, Nested conditionals, The return statement, Recursion, Stack diagrams for recursive functions, Multiple assignment, The while statement, Tables, Two-dimensional tables Strings and Lists.

Strings and Lists: String as a compound data type, Length, Traversal and the for loop, String slices, String comparison, A find function, Looping and counting, List values, Accessing elements, List length, List membership, Lists and for loops, List operations, List deletion. Cloning lists, Nested lists

Object Oriented Programming: Introduction to Classes, Objects and Methods, Standard Libraries.

Fundamental and Programming in Python Lab

Practical: 60 lectures

Basic Assignments on Office, Excel and Power point

1. 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.
2. 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) and tan(x).
3. Write a program that reads an integer value and prints —leap year‖ or —not a leap year‖.
4. 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
   ```
   *
   **
   ***
   ****
   *****
   ```
5. Write a function that takes an integer ‗n‗ as input and calculates the value

   of 1 + 1/1! + 1/2! + 1/3! + … + 1/n
6. Write a function that takes an integer input and calculates the factorial of that number.
7. Write a function that takes a string input and checks if it‗s a palindrome or not.
8. Write a list function to convert a string into a list, as in list (‗abc‗) gives [a, b, c].
9. Write a program to generate Fibonacci series.
10. Write a program to check whether the input number is even or odd.

11. Write a program to compare three numbers and print the largest one.
12. Write a program to print factors of a given number.
13. Write a method to calculate GCD of two numbers.
14. Write a program to create Stack Class and implement all its methods. (Use Lists).
15. Write a program to create Queue Class and implement all its methods. (Use Lists)
16. Write a program to implement linear and binary search on lists.
17. Write a program to sort a list using insertion sort and bubble sort and selection sort.

Data Structure using Python

Practical :60 lectures

Module 01:

Informal introduction to programming, algorithms and data structures ,variables, operations, control flow - assignments, conditionals, loops, functions.

Module 02

Scope in Python: local, global, nonlocal names , Nested functions | Data structures: Arrays ,stack, queue , Heaps.

Module 03

Python: types, expressions, strings, lists, tuples , Python memory model: names, mutable and immutable values , List operations: slices etc - Binary search | Inductive function definitions: numerical and structural induction ,Elementary inductive sorting: selection and insertion sort ,In-place sorting, | Merge sort ,Quicksort , Stable sorting.

Module 04

Basic algorithmic analysis: input size, asymptotic notations, complexity.

Module 05

Dictionaries | More on Python functions: optional arguments, default values | Passing functions as arguments | Higher order functions on lists: map, iter, list comprehension.

Module 06

Abstract datatypes | Classes and objects in Python | "Linked" lists: Find, insert, delete | Binary search trees: Find, insert, delete | Height-balanced binary search trees.

Module 07

Exception handling | Basic input/output | Handling files | String processing.

References :
3. T. Budd, Exploring Python, TMH, 1st Ed, 2011
4. How to think like a computer scientist : learning with Python / Allen Downey, Jeffrey Elkner, Chris Meyers. 1st Edition – Freely available online.2012
1. http://docs.python.org/3/tutorial/index.html
2. http://interactivepython.org/courselib/static/pythonds

Data Structure using Python Lab:

Practical 60 Lectures

Write Python programs for implementing the following searching techniques.

a.Linear search

b.Binary search

c.Fibonacci search

Write Python programs for implementing the following sorting techniques to arrange a list of integers in ascending order.

a.Bubble sort

b.Insertion sort

c.Selection sort

Write Python programs for implementing the following sorting techniques to arrange a list of integers in ascending order.

a.Quick sort

b.Merge sort

Write Python programs to

a.Designand implement Stack and its operations using List.

b.Design and implement Queue and its operations using List.

Write Python programs for the following:

a.Uses Stack operations to convert in fix expression into postfixexpression.

b.Uses Stack operations for evaluating the post fix expression.

a. Write Python programs for the following operations on Single Linked List.

(i)Creation(ii)insertion(iii)deletion(iv)traversal

b. To store a polynomial expression in memory using single linkedlist.

Write a Pythonprogram to perform the following:

a.Create a binary search tree.

b.Traverse the above binary search tree recursively in pre-order, post-order and in-order.

c.Count the number of nodes in the binary search tree

LIST OF REFERENCE BOOKS:

1.Y Daniel Liang, ―Introduction to Programming using Python‖, Pearson.

2.Benjamin Baka, David Julian, ―Python Data Structures and Algorithms‖, Packt Publishers,2017.

3.Rance D. Necaise, ―Data Structures and Algorithms using Python‖, Wiley Student Edition.

4.Martin Jones, ―Python for Complete Beginners‖, 2015.

5.Zed A. Shaw, ―Learn Python the Hard Way: a very simple introduction to the terrifyingly beautiful world of computers and code‖, 3e, Addison-Wesley, 2014.

6.Hemant Jain, ―Problem Solving in Data Structures and Algorithms using Python: programming interview guide‖, 2016.