

**Syllabus of**  
**B. Sc. (Honours) in Computer Science**  
**(CHOICE BASED CREDIT SYSTEM)**



**This is approved in the Academic Council held on 8/11/2019**

## 1. Outline of courses in Choice Based Credit System

**1.1 Core Course:** A course, which should compulsorily be studied by a candidate as a core requirement is termed as a Core course.

**1.2 Elective Course:** Generally a course which can be chosen from a pool of courses and which may be very specific or specialized or advanced or supportive to the discipline/subject of study or which provides an extended scope or which enables an exposure to some other discipline/subject/domain or nurtures the candidate's proficiency/skill is called an Elective Course.

**1.2.1 Discipline Specific Elective (DSE) Course:** Elective courses that are offered by the main discipline/subject of study are referred to as Discipline Specific Electives.

**1.2.2 Dissertation/Project:** An elective course designed to acquire special/advanced knowledge in the subject. It is considered as a special course involving application of knowledge in solving/analyzing/exploring a real life situation/difficult problem. A candidate studies such a course on his own with an advisory support by a teacher/faculty member. The work done has to be submitted as a dissertation at the end of the semester.

**1.2.3 Generic Elective (GE) Course:** An elective course chosen generally from another discipline/subject, with an intention to seek exposure is called a Generic Elective.

### 1.3 Ability Enhancement Courses (AEC)

The Ability Enhancement (AE) Courses are the course that leads to Knowledge enhancement. These are of two types.

**1.3.1 AE Compulsory Course (AECC):** Environmental Studies, English Communication/MIL Communication.

**1.3.2 AE Elective Course (AEEC):** AEEC courses are value-based and/or skill-based and are aimed at providing hands-on-training, competencies, skills, etc. These courses are to be chosen from a pool of courses designed to provide value-based and/or skill-based instruction.

## 2. BACHELOR OF SCIENCE (HONOURS) COMPUTER SCIENCE Program Details

### 2.1 Prerequisite

Students, who want to take admission in B.Sc. (Honours) Computer Science Program, need to secure minimum mark for passing in Mathematics at **Class XII/ Higher Secondary/ any equivalent** examination.

### 2.2 Program Objectives

Students who choose B.Sc. (Honours) Computer Science Program, will develop the ability to think critically, logically, analytically and to use and apply current technical concepts and practices in the core development of solutions in the form of Information Technology.

The knowledge and skills gained with a degree in Computer Science prepare graduates for a broad range of jobs in Education sector, Research field, Government sector, Business sector and Industry.

The program covers the various essential concepts in Computer Science. These are included as core course like Structured Foundation of Computer Fundamentals, Computing Methods,

Data Structure, Software Engineering, Programming Concepts in various languages (C, C++, Java, Visual Basic etc.), Design and Analysis of Algorithm, Theory of Computation, System Programming, Computer Networking, System Administration, Operating System, Computer Architecture, Microprocessor, PHP programming, Numerical Methods, Computer Graphics and Database Management System.

An exceptionally broad range of topics covering current trends and technologies in Computer Science like - Programming in Python, Information Security and Cyber Laws, Data Mining, R-Programming, E-commerce, Data Sciences, Internet Technologies, Artificial Intelligence, Android Programming, UNIX/ LINUX programming etc are included in the course. Hands on sessions in Computer Lab using various Programming languages and tools will enable students to deal with real life problems which will lead to better understanding of the topics and will also widen the horizon of students' self-experience.

### 2.3 Choosing of Other Discipline papers

Students, who choose B.Sc. (Honours) Computer Science Program, need to select any one discipline from the following list of Other Disciplines for their four GE papers GE1 to GE4.

#### Other Discipline (Four papers of anyone discipline) – GE 1 to GE 4

1. Mathematics
2. Statistics
3. Operational Research
4. Physics
5. Electronics
6. Commerce
7. Economics

### 2.4 Program Learning Outcomes:

Completion of B.Sc. (Honours) Computer Science Program shall enable a student

- i) To communicate technical information both orally and in writing
- ii) Apply the knowledge gained in core courses to a broad range of advanced topics in Computer Science, to learn and develop sophisticated technical products independently.
- iii) To design, implement and evaluate computer-based system, process, component, or program to meet desired needs by critical understanding, analysis and synthesis.
- iv) Identify applications of Computer Science in other fields in the real world to enhance the career prospects
- v) Realize the requirement of lifelong learning through continued education and research.
- vi) Use the concepts of best practices and standards to develop user interactive and abstract application
- vii) Understand the professional, ethical, legal, security, social issues and responsibilities

### 2.5 Program Structure:

The B.Sc. (Honours) Computer Science Program is a three-year course divided into six- semesters. A student is required to complete 148 credits for the completion of course and the award of degree.

		Semester	Semester
<b>Part – I</b>	First Year	Semester I : 22	Semester II : 22
<b>Part – II</b>	Second Year	Semester III: 28	Semester IV: 28
<b>Part – III</b>	Third Year	Semester V:24	Semester VI:24

## 2.6 Program Implementation Requirement:

The B.Sc. (Honours) Computer Science Program is a three-year course divided into six- semesters. For proper implementation of the UGCBCS program the following infrastructure are necessary:

- a) Sufficient lab facilities with computers and software
- b) At least 6 full time teaching faculties.

## 2.7 Instruction for questions paper setter:

Question Paper setter should set from the prescribed text books, mentioned in the syllabus.

## 3. Credit allocation (B.Sc. Honours)

Course	Credits	Credits
	Theory + Practical	Theory + Tutorial
I. Core Course (6 credits) (14 papers)	$14 \times 4 = 56$	$14 \times 5 = 70$
Core Course Practical / Tutorial (14 Papers)	$14 \times 2 = 28$	$14 \times 1 = 14$
II. Elective Course (6 credits) (8 Papers)		
A.1. Discipline Specific Elective (4 Papers)	$4 \times 4 = 16$	$4 \times 5 = 20$
A.2. Discipline Specific Elective Practical/ Tutorial (4 Papers)	$4 \times 2 = 8$	$4 \times 1 = 4$
B.1. Generic Elective/ Interdisciplinary (4 Papers)	$4 \times 4 = 16$	$4 \times 5 = 20$
B.2. Generic Elective Practical/ Tutorial (4 Papers)	$4 \times 2 = 8$	$4 \times 1 = 4$
Optional dissertation or project work in place of one Discipline Specific Elective paper (6 credits) in 5 <sup>th</sup> semester		
III Ability Enhancement Courses		
1. Ability Enhancement Compulsory Courses (AECC) (2 Papers of 4 credit each)	$2 \times 4 = 8$	$2 \times 4 = 8$
Environmental Science		
English Communication		
2. Skill Enhancement Courses (SEC) (Minimum 2) (2 Papers of 4 credit each)	$2 \times 4 = 8$	$2 \times 4 = 8$
Total credit	148	148

## CBCS Course Structure for B.Sc. (Honours) Computer Science Program

### SEMESTER WISE PLACEMENT OF THE COURSES

Semester	CORE COURSE (14)	Ability Enhancement Compulsory Course(AECC) (2)	Skill Enhancement Course (SEC) (2)	Elective: Discipline Specific DSE (4)	Elective: Generic(GE) (4)
<b>I</b>	CSC-HC-1016 Programming Fundamentals using C/C++	ENG-AE-1014			GE-1
	CSC-HC-1026 Computer System Architecture				
<b>II</b>	CSC-HC-2016 Programming in JAVA	ENV-AE-2014			GE-2
	CSC-HC-2026 Discrete Structures				
<b>III</b>	CSC-HC-3016 Data Structures		SEC -1		GE-3
	CSC-HC-3026 Operating System				
	CSC-HC-3036 Computer Networks				
<b>IV</b>	CSC-HC-4016 Design and Analysis of Algorithms		SEC -2		GE-4
	CSC-HC-4026 Software Engineering				
	CSC-HC-4036 Database Management System				
<b>V</b>	CSC-HC-5016 Internet Technologies			DSE-1	
	CSC-HC-5026 Theory of Computation			DSE -2	
<b>VI</b>	CSC-HC-6016 Artificial Intelligence			DSE -3	
	CSC-HC-6026 Computer Graphics			DSE -4	

#### Discipline Specific Electives (DSE)

##### DSE 1 (choose one)

- (i) CSC-HE-5016: Microprocessor
- (ii) CSC-HE-5026: Numerical Methods

##### DSE 2

- (i) CSC-HE-5036: Project Work / Dissertation

**DSE 3, DSE 4 (choose two)**

- (i) CSC-HE-6016: Network Programming
- (ii) CSC-HE-6026: Introduction to Data Sciences
- (iii) CSC-HE-6036: System Programming
- (iv) CSC-HE-6046: Data Mining

**Skill Enhancement Course (SEC)**

**SEC 1 (choose one)**

- (i) CSC-SE-3014: UNIX/LINUX Programming
- (ii) CSC-SE-3024: Programming in Python
- (iii) CSC-SE-3034: HTML Programming

**SEC 2 (choose one)**

- (i) CSC-SE-4014: Android Programming
- (ii) CSC-SE-4024: PHP Programming
- (iii) CSC-SE-4034: R Programming

**Generic Elective (GE)**

**(N.B: The list of GE papers given below are for the students of B.Sc. (Honours) from any discipline other than Computer Science (if applicable), who want to take Computer Science as their Generic Elective course.**

**GE 1 (choose one)**

- (i) CSC-HG-1016: Problem Solving using Computer (CSC-RC-1016)
- (ii) CSC-HG-1026: Multimedia and Applications
- (iii) CSC-HG-1036: Fundamentals of Computer Applications ( TCA-RC-1016)
- (iv) CSC-HG-1046: Fundamentals of Information Technology (TIT-RC-1016)

**GE 2 (Choose one)**

- (i) CSC-HG-2016: Programming in C++
- (ii) CSC-HG-2026: Database Management System (CSC-RC-2016)
- (iii) CSC-HG-2036: Introduction to Programming in C ( TCA-RC-2016)/ (TIT-RC-2016)

**GE 3 (choose one)**

- (i) CSC-HG-3016: Programming in Python
- (ii) CSC-HG-3026: Computer Networks and Internet Technologies
- (iii) CSC-HG-3036: Operating Systems(CSC-RC-3016)/( TCA-RC-3016)
- (iv) CSC-HG-3046: ICT Hardware (TIT-RC-3016)

**GE 4 (choose one)**

- (i) CSC-HG-4016: Programming in Visual Basic / Gambas
- (ii) CSC-HG-4026: Information Security and Cyber Laws
- (iii) CSC-HG-4036: Web and E-Commerce Technologies
- (iv) CSC-HG-4046: Computer System Architecture (CSC-RC-4016)
- (v) CSC-HG-4056: Introduction to DBMS ( TCA-RC-4016)/ (TIT-RC-4016)

# Detailed Syllabus

## CSC-HC-1016: Programming Fundamentals using C/C++

4 Lectures, 4 Practical, Credits 6 (4+2)

Theory: 60 Lectures, Practical: 60 Lectures

End Semester Marks:

Theory: 60 Marks, Practical: 20 Marks

Internal Marks:

Sessional: 10 Marks, Practical: 6 Marks, Attendance: 4 Marks

### **UNIT 1: Introduction to C and C++ (1 Lectures)**

History of C and C++, Overview of Procedural Programming and Object-Oriented Programming

### **UNIT 2: Data Types, Variables, Constants, Operators and Basic I/O (5 Lectures)**

Data Types, Declaration, Definition and Initialization of Variables, Scope of Variables, Constants, Keywords, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Character I/O (getc, getchar, putc, putchar), Formatted and Console I/O (printf(), scanf(), cin, cout), Basic Header Files (stdio.h, iostream.h, conio.h etc).

### **UNIT 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)

### **UNIT 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

### **UNIT 5: Derived Data Types (Structures and Unions) (4 Lectures)**

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

**UNIT 6: Pointers and References in C++****(8 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

**UNIT 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

**UNIT 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

**UNIT 9: Using Classes in C++****(6 Lectures)**

Principles of Object-Oriented Programming, Class, Object, Class Variables & Functions, Specifying the Protected and Private Access, Objects as parameters, Constructors, Copy Constructor, Constructor Overloading

**UNIT 10: Overview of Function Overloading and Operator Overloading****(6 Lectures)**

Overview of Polymorphism, Need of Overloading functions and operators, Function overloading in classes, Overloading Operators (including assignment operators, unary operators), Overview of Template classes and their use.

**UNIT 11: Inheritance and Exception Handling****(8 Lectures)**

Introduction to Inheritance, Types of Inheritance, Virtual base class, Basics Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions, Restricting exceptions, Rethrowing exceptions.

**Practical / Lab work to be performed**

**(N.B: Student has to perform any fifteen of the following experiments)**

1. Write a program to print the sum and product of digits of an integer.
2. Write a program to reverse a number.
3. Write a program to compute the sum of the first n terms of the following series  
$$S = 1 + 1/2 + 1/3 + 1/4 + \dots$$
4. Write a program to compute the sum of the first n terms of the following series  
$$S = 1 - 2 + 3 - 4 + 5 - \dots$$
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. Write a program to compute the factors of a given number.



8. Write a macro that swaps two numbers. Write a program to use it.
9. Write a program to print a triangle of stars as follows (take number of lines from user):

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

10. Write a program to perform following actions on an array entered by the user:
  - i) Print the even-valued elements
  - ii) Print the odd-valued elements
  - iii) Calculate and print the sum and average of the elements of array
  - iv) Print the maximum and minimum element of array
  - v) Remove the duplicates from the array
  - vi) Print the array in reverse orderThe 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. Write a program 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:
  - i) Show address of each character in string
  - ii) Concatenate two strings without using strcat function.
  - iii) Concatenate two strings using strcat function.
  - iv) Compare two strings
  - v) Calculate length of the string (use pointers)
  - vi) Convert all lowercase characters to uppercase
  - vii) Convert all uppercase characters to lowercase
  - viii) Calculate number of vowels
  - ix) Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. Write a program to display Fibonacci series (i) using recursion, (ii) using iteration
19. Write a program to calculate Factorial of a number (i) using recursion, (ii) using iteration
20. Write a program 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):
  - i) Sum ii) Difference iii) Product iv) 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:
    - i) Calculate surface Area
    - ii) Calculate Volume
    - iii) Increment, Overload ++ operator (both prefix & postfix)
    - iv) Decrement, Overload -- operator (both prefix & postfix)
    - v) Overload operator == (to check equality of two boxes), as a friend function
    - vi) Overload Assignment operator
    - vii) Check if it is a Cube or cuboidWrite 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.

## Reference Books

1. HerbtzSchildt, "C++: The Complete Reference", Fourth Edition, McGraw Hill.2003
2. BjarneStroustrup, "The C++ Programming Language", 4<sup>th</sup> Edition, Addison-Wesley , 2013.
3. BjarneStroustrup, "Programming -- Principles and Practice using C++", 2<sup>nd</sup> 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.
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 , 2000.
8. Scott Meyers, "Effective C++", 3rd Edition, Published by Addison-Wesley, 2005.
9. Harry, H. Chaudhary, "Head First C++ Programming: The Definitive Beginner's Guide", 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-Wesley, 5th Edition, 2012

## **CSC-HC-1026: Computer System Architecture**

4 Lectures, 4 Practical, Credits 6 (4+2)

Theory: 60 Lectures, Practical: 60 Lectures

End Semester Marks:

Theory: 60 Marks, Practical: 20 Marks

Internal Marks:

Sessional: 10 Marks, Practical: 6 Marks, Attendance: 4 Marks

### **UNIT 1: Introduction (12 lectures)**

Logic gates, Boolean algebra, Combinational Circuits, Circuit Simplification, Flip-Flops and Sequential Circuits, Encoders, Decoders, Multiplexers, De-multiplexers, Registers, Counters and Memory Units.

### **UNIT 2: Data Representation and Basic Computer Arithmetic (8 lectures)**

Number systems, Complements, Fixed and Floating point representation, Representation of signed numbers, Addition, Subtraction, Magnitude comparison, Character representation.

### **UNIT 3: Basic Computer Organization and Design (18 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, Design of control unit, Design of ALU.

### **UNIT 4: Central Processing Unit (12 lectures)**

Register organization, Arithmetic, logical and shift micro-operations, Stack organization, Micro programmed control, Instruction formats, Addressing modes, Instruction codes, Basic concepts of machine language, Assembly language.

### **UNIT 5: Memory Organization (5 lectures)**

Basic concepts of RAM, ROM, Secondary memory, Cache memory, Associative memory, Mapping.

### **UNIT 6: Input-Output Organization (5 lectures)**

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

### **Practical / Lab work to be performed:**

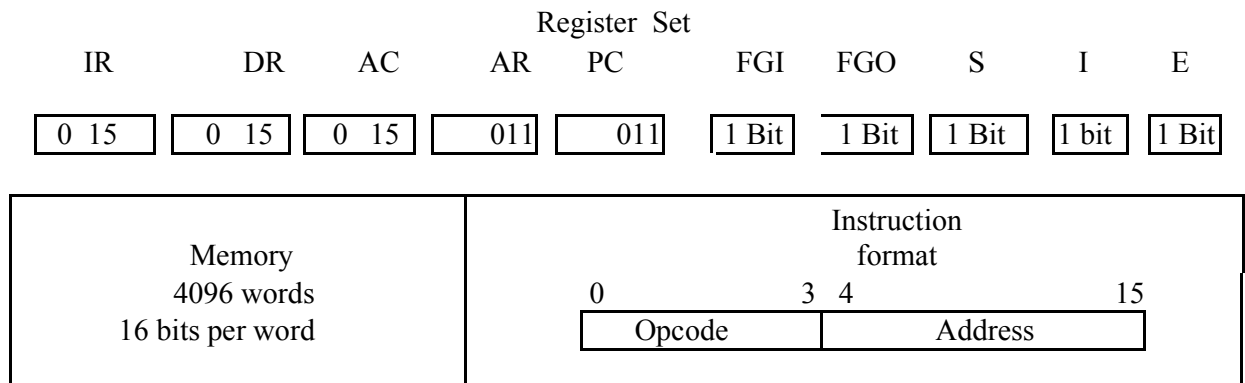
(N.B: Student has to perform **any five** of the following experiments)

1. Construct a 2-to-4 line decoder with only NOR gates.
2. Construct a 16-to-1 line multiplexer with two 8-to-1 line multiplexers and one 2-to-1 line multiplexer.
3. Construct an Arithmetic Unit capable of performing 4-bit subtraction and Addition using 2's complement method. Use Parallel Adders and other necessary logic gates.

- Construct a logical Unit using logic gates capable of performing 4-bit, Bitwise OR, AND, XOR and inversion.
- Construct an 4-bit ALU unit which can perform the following operation;

Selection			Function
S0	S1	S2	
0	0	0	ADD
0	0	1	SUBTRACT
0	1	0	INCREMENT
0	1	1	DECREMENT
1	0	0	MULTIPLY BY 2
1	0	1	DIVIDE BY 2
1	1	0	BITWISE AND
1	1	1	BITWISE OR

- Construct a 4 bit ring counter.
- Create a machine based on the following architecture:



Basic Computer Instructions					
Memory Reference		Register Reference		Input-Output	
Symbol	Hex	Symbol	Hex	Symbol	Hex
AND	0xxx	CLA	E800	INP	F80 0
STA_I	7xxx	SZE	E002		F40
BUN_I	9xxx	HLT	E001		
BSA_I	Bxxx				
ISZ_I	Dxxx				

## Reference Books

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

# **CSC-HC-2016: Programming in JAVA**

4 Lectures, 4 Practical, Credits 6 (4+2)  
Theory: 60 Lectures, Practical: 60 Lectures

## **UNIT 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)

## **UNIT 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.

## **UNIT 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.

## **UNIT 4: 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.

## **UNIT 5: 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 use JDBC.

## **UNIT 6: Applets and Event Handling (15 Lectures)**

Java Applets: Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds, Event Handling Mechanisms, Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT

controls, Swing components of Java Foundation Classes such as labels, buttons, text fields, layout managers,

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

### **Practical / Lab work to be performed**

**(N.B:** Student has to perform **any fifteen** of the following experiments)

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of .length in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and StringBuffer classes like setCharAt(), setLength(), append(), insert(), concat() and equals().
9. Write a program to create a —distance class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
10. Modify the —distance class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions (from lower to higher data type)
12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
13. Write a program to show the use of static functions and to pass variable length arguments in a function.
14. Write a program to demonstrate the concept of boxing and unboxing.
15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate fibonacci series is given in a different file belonging to the same package.
17. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
18. Write a program —DivideByZero that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
20. Write a program to create your own exception types to handle situation specific to your

- application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
21. Write a program to demonstrate priorities among multiple threads.
  22. Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).
  22. Write a program to create URL object, create a URLConnection using the openConnection() method and then use it examine the different components of the URL and content.
  23. 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.
  24. 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.
  25. Write a program to get the URL/location of code (i.e. java code) and document(i.e. html file).
  26. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().
  27. 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.

## 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, Gilad Bracha, 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, Gary Cornell, "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. David J. Eck, "Introduction to Programming Using Java", Published by CreateSpace Independent Publishing Platform, 2009.
10. John R. Hubbard, "Programming with JAVA", Schaum's Series, 2nd Edition, 2004.

# CSC-HC-2026: Discrete Structures

5 Lectures, 1 Tutorial, Credits 6 (5+1)  
Theory: 60 Lectures, Tutorial: 15 Lectures

## UNIT 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.

## UNIT 2: Growth of Functions

(8 Lectures)

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

## UNIT 3: Recurrences

(10 Lectures)

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

## UNIT 4: Graph Theory

(15 Lectures)

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

## UNIT 5: Propositional Logic

(12 Lectures)

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

## Reference Books

1. C.L. Liu , D.P. Mahopatra, Elements of Discrete mathematics, 2<sup>nd</sup> Edition , Tata McGraw Hill, 1985,
2. Kenneth Rosen, Discrete Mathematics and Its Applications, Sixth Edition ,McGraw Hill 2006
3. T.H. Cormen, 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



## **CSC-HC-3016: Data Structures**

4 Lectures, 4 Practical, Credits 6 (4+2)  
Theory: 60 Lectures Practical: 60 Lectures

- UNIT 1: Arrays** **(5 Lectures)**  
Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation), Insertion and Deletion of elements from an Array and Amount of Work Done
- UNIT 2: Stacks** **(5 Lectures)**  
Implementing single / multiple stacks in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another, Applications of stack, Limitations of Array representation of stack
- UNIT 3: Linked Lists** **(10 Lectures)**  
Singly, Doubly and Circular Lists (Array and Linked representation), Insertion, Deletion and Searching in different Linked Lists (Different cases like Inserting/ Deleting at the end, at the starting etc), Advantages and disadvantages of linked lists over array structure, Normal and Circular representation of Stack in Lists, Self Organizing Lists, Skip Lists
- UNIT 4: Queues** **(5 Lectures)**  
Array and Linked representation of Queue, De-queue, Priority Queues
- UNIT 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)
- UNIT 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 (Definitions), Concepts of Height Balanced Trees
- UNIT 7: Searching and Sorting** **(5 Lectures)**  
Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Comparison of Sorting Techniques
- UNIT 8: Hashing** **(5 Lectures)**  
Introduction to Hashing, Deleting from Hash Table, Efficiency of Rehash Methods, Hash Table Reordering, Resolving collision by Open Addressing, Coalesced Hashing, Separate Chaining, Dynamic and Extendible Hashing, Choosing a Hash Function, Perfect Hashing Function

## Practical / Lab work to be performed

(N.B: Student has to perform **any ten** of the following experiments)

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. Write a program using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. Write a program to scan a polynomial using linked list and add two polynomial.
11. Write a program to calculate factorial and to compute the factors of a given no.  
(i) using recursion, (ii) using iteration
12. Write a program to display Fibonacci series (i) using recursion, (ii) using iteration
13. Write a program to calculate GCD of 2 number (i) with recursion (ii) without recursion
14. Write a program to create a Binary Search Tree and include following operations in tree:
  - (a) Insertion (Recursive and Iterative Implementation)
  - (b) Deletion by copying
  - (c) Deletion by Merging
  - (d) Search a no. in BST
  - (e) Display its preorder, postorder and inorder traversals Recursively
  - (f) Display its preorder, postorder and inorder traversals Iteratively
  - (g) Display its level-by-level traversals
  - (h) Count the non-leaf nodes and leaf nodes
  - (i) Display height of tree
  - (j) Create a mirror image of tree
  - (k) Check whether two BSTs are equal or not
15. Write a program to convert the Sparse Matrix into non-zero form and vice-versa.
16. Write a program to reverse the order of the elements in the stack using additional stack.
17. Write a program to reverse the order of the elements in the stack using additional Queue.
18. Write a program to implement Diagonal Matrix using one-dimensional array.
19. Write a program to implement Lower Triangular Matrix using one-dimensional array.
20. Write a program to implement Upper Triangular Matrix using one-dimensional array.

21. Write a program to implement Symmetric Matrix using one-dimensional array.
22. Write a program to create a Threaded Binary Tree as per inorder traversal, and implement operations like finding the successor / predecessor of an element, insert an element, inorder traversal.
23. Write a program to implement various operations on AVL Tree.

## 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.

# CSC-HC-3026: Operating System

4 Lectures, 4 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

**UNIT 1: Introduction** **(11 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

**UNIT 2: Operating System Organization** **(9 Lectures)**  
Processor and user modes, kernels, system calls and system programs.

**UNIT 3: Process Management** **(22 Lectures)**  
System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries, Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms; concurrent and processes, critical section, semaphores, methods for inter-process communication; deadlocks, Physical and virtual address space, memory allocation strategies –fixed and variable partitions, paging, segmentation, virtual memory

**UNIT 4: File and I/O Management** **(12 Lectures)**  
Directory structure, File operations, File allocation methods, Device management

**UNIT 5: Protection and Security** **(6 Lectures)**  
Policy mechanism, Authentication, Internal access Authorization.

## **Practical / Lab work to be performed**

(N.B: Student has to perform **any six** of the following experiments)

1. Write a program (using *fork()* and/or *exec()* commands) where parent and child execute:
  - a) same program, same code.
  - b) same program, different code.
  - c) before terminating, the parent waits for the child to finish its task.
2. Write a program to report behavior of Linux kernel including kernel version, CPU type and model. (CPU information)
3. Write a program to report behavior 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.

## Reference Books

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8<sup>th</sup> Edition, John Wiley Publications 2008.
2. A.S. Tanenbaum, Modern Operating Systems, 3<sup>rd</sup> Edition, Pearson Education 2007.
3. G. Nutt, Operating Systems: A Modern Perspective, 2<sup>nd</sup> Edition Pearson Education 1997.
4. W. Stallings, Operating Systems, Internals & Design Principles , 5<sup>th</sup> Edition, Prentice Hall of India. 2008.
5. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

# **CSC-HC-3036: Computer Networks**

4 Lectures, 4 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

- UNIT 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
- UNIT 2: Data Communication Fundamentals and Techniques (10 Lectures)**  
Analog and digital signal, data-rate limits, digital to digital line encoding schemes, pulse code modulation, parallel and serial transmission, digital to analog modulation, multiplexing techniques- FDM, TDM, transmission media
- UNIT 3: Networks Switching Techniques and Access mechanisms (10 Lectures)**  
Circuit switching, packet switching- connectionless datagram switching, connection-oriented virtual circuit switching, dial-up modems, digital subscriber line, cable TV for data transfer
- UNIT 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.
- UNIT 5: Multiple Access Protocol and Networks (5 Lectures)**  
CSMA/CD protocols, Ethernet LANS, connecting LAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways
- UNIT 6: Networks Layer Functions and Protocols (6 Lectures)**  
Routing, Routing algorithms, Network Layer Protocol of Internet- IP protocol, Internet control protocols
- UNIT 7: Transport Layer Functions and Protocols (6 Lectures)**  
Transport Services- Error and Flow control, Connection establishment and release- Three Way Handshake
- UNIT 8: Overview of Application layer protocol (5 Lectures)**  
Overview of DNS protocol, Overview of WWW & HTTP protocol

## **Practical / Lab work to be performed**

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate and implement stop and wait protocol for noisy channel.
3. Simulate and implement go back n sliding window protocol.
4. Simulate and implement selective repeat sliding window protocol.
5. Simulate and implement distance vector routing algorithm
6. Simulate and implement Dijkstra algorithm for shortest path routing.

## **Reference Books**

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM ,2007.
2. A. S. Tanenbaum: Computer Networks, Fourth edition, PHI , 2002

# CSC-HC-4016: Design and Analysis of Algorithms

4 Lectures 4 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Introduction** **(6 Lectures)**

Basic Design and Analysis techniques of Algorithms, Correctness of Algorithm.

## **UNIT 2: Algorithm Design Techniques** **(10 Lectures)**

Iterative techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.

## **UNIT 3: Sorting and Searching Techniques** **(20 Lectures)**

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

## **UNIT 4: Balanced Trees** **(9 Lectures)**

Height of a Red-Black tree, Rotations - Left Rotations, Right Rotations, Insertion and Deletion in Red-Black trees.

## **UNIT 5: Graphs** **(9 Lectures)**

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

## **UNIT 6: String Processing** **(6 Lectures)**

String Matching, KMP Technique

### **Practical / Lab work to be performed**

(N.B: Student has to perform **any five** of the following experiments)

1. Implement Insertion Sort (The program should report the number of comparisons)
2. Implement Merge Sort(The program should report the number of comparisons)
3. Implement Heap Sort(The program should report the number of comparisons)
4. Implement Randomized Quick sort (The program should report the number of comparisons)
5. Implement Radix Sort
6. 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  $n \log n$ .

### **Reference 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

# CSC-HC-4026: Software Engineering

4 Lectures, 4 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

- UNIT 1: Introduction (10 Lectures)**  
The Evolving Role of Software, Software Characteristics, Software v/s Programs, Emergence of Software Engineering, Notable Changes in Software Development Practices, Software Development Life Cycle Models (SDLC): *Build and Fix, V-Model, Waterfall models, Prototyping Model, Incremental Model, Evolutionary model, Spiral Model*
- UNIT 2: Requirement Analysis (4 Lectures)**  
Software Requirement Analysis and Gathering, Different ways of Requirement Gathering Process, Requirement Analysis and Modeling Techniques, Need for SRS, Components of SRS, Characteristics of good and bad SRS
- UNIT 3: Software Project Management (14 Lectures)**  
Goal of Software Project management, Organization of SPMP (Software Project Management Plan) document, Estimation in Project Planning Process, Lines of Code (LOC), Function Point (FP) metric, Project Estimation Techniques: (*Empirical Estimation, Heuristics Estimation*), Expert Judgment, Delphi Cost Estimation, COCOMO models, Project Scheduling, Work breakdown structure, Activity Network Representation, Critical Path Method, Staffing level estimation, Norden's work, Putnam's work
- UNIT 4: Risk Management (4 Lectures)**  
Software Risks, Risk Identification, Types of Risks, Risk Assessment, Risk Refinement, Risk Mitigation, Monitoring and Management Plan (RMMM) Plan
- UNIT 5: Quality Management (8 Lectures)**  
Quality Concepts, Evolution of Quality Systems, Software Quality Assurance (QA), Total Quality Management (TQM), ISO 9000, How to get ISO 9000 Certification, Shortcomings of ISO 9000, SEI-Capability Maturity Model (CMM) and Integration
- UNIT 6: Design Engineering (10 Lectures)**  
Design Concepts, Cohesion and Coupling, Classification of Cohesion and Coupling, Function Oriented design vs Structured Design, Elements of a Data Flow Diagram (DFDs), properties of a DFD, Balancing DFD, Synchronous and Asynchronous DFD, Structured design (SD) concepts, Basic elements used to design a SD, Design Review
- UNIT 7: Testing Strategies & Tactics (10 Lectures)**  
Software Testing Fundamentals, Strategic Approach to Software Testing, Verification and validation process, Unit Testing, Integration Testing :*Top Down approach* and *Bottom up approach*, System testing- Black-Box Testing: *Equivalent partitioning* and *boundary value analysis*, White-Box Testing and their type: *Statement Coverage*,

*Branch Coverage, Multiple Condition Coverage and Basis Path Testing (Path Coverage), Control Flow Graph (CFG), McCabe's Cyclomatic Complexity metric,*

## **Practical / Lab work to be performed**

**(N.B: Student has to perform *any eight* of the following experiments)**

**Experiment-1:** Consider a Software project to do the followings: *Define problem Statement, Needs of the project, Identify the suitable life cycle model to use, possible software crisis, Strategies and techniques used for Requirement analysis and Gathering.*

**Experiment-2:** Consider a project and design the suitable SRS document for the same.

**Experiment-3:** Execute the LOC size estimation metric for a given problem.

**Experiment-4:** Apply the Function Point (FP) size estimation metric for a given problem with suitable technical constrains.

**Experiment-5:** Consider a project to implement COCOMO model to estimate effort of the project

**Experiment-6:** Apply the Norden's model of staffing level estimation for a given problem definition.

**Experiment-7:** Consider a project to implement Putnam's model of staffing level estimation.

**Experiment-8:** Consider a problem definition to apply equivalent class partition technique to develop test cases.

**Experiment-9:** Apply Boundary value analysis method to develop test cases for a project.

**Experiment-10:** Design test suits for a given code segment applying statement Coverage testing technique.

**Experiment-11:** Design test suits for a given code segment applying Branch Coverage testing technique.

**Experiment-12:** Apply the Basis path testing for a given code segment of a project.

**Experiment-13:** Design the DFD for a given project. (N.B: *using Software tools*)

**Experiment-14:** Design the Structure Chart for a given project. (N.B: *using Software tools*)

**Sample List of Projects:** (N.B: one or more projects from the given list can be considered as a basis for the laboratory work)

1. **Criminal Record Management:** 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:** Maintain information about 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
11. Library Management System
12. Attendance Management System

## **Reference 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 (2nd Edition), Naros Publishing House, 2003.
3. K.K. Aggarwal and Y. Singh, Software Engineering ( 2nd Edition), New Age International Publishers, 2008.
4. I. Sommerville, Software Engineering (8th edition), Addison Wesley, 2006.
5. D. Bell, Software Engineering for Students (4th Edition), Addison-Wesley, 2005.
6. R. Mall, Fundamentals of Software Engineering (2nd Edition), Prentice-Hall of India, 2004.

# CSC-HC-4036: Database Management System

4 Lectures, 4 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

## UNIT 1: Introduction (6 Lectures)

Definition of Database, Traditional File Approach vs. DBMS approach, Characteristics of the Database Approach, DBMS users, Role of a DBA, Advantage of using DBMS, Database System architecture, ANSI/SPARC 3 level architecture, Data independence

## UNIT 2: Entity Relationship(ER) Modeling (8 Lectures)

DFD, ER model as a tool for conceptual design-entities, Entity types, Relationships, Constraints,

## UNIT 3: Relation data model (20 Lectures)

Relational model concepts, Relational Constraints: Entity integrity, Referential integrity, Relational algebra (Select , Project, Cross ,Product , theta join, equi join, natural join, outer join ), Set Operation DDL, DML, SQL queries: SQL constructs (Select .. From... Where... Group by ..... Having... Order by....), Insert, Delete, Update, View, Definition and use, nested quires, Constraints considers(NOT NULL , UNIQUE, Check Primary key. Foreign key)

## UNIT 4: Database design (15 Lectures)

Mapping ER/EER model to relational database, different types of keys, functional dependencies, Lossless decomposition, Normal forms (upto BCNF), Database Connectivity using JDBC.

## UNIT 5: Transaction processing (3 Lectures)

ACID properties, Concurrency Control

## UNIT 6: File Structure and Indexing (8 Lectures)

File Structure: Placing file records on disks: Records and its Types, Files, Fixed length records and Variable length records, Record Blocking, Allocating file blocks on disks, Operation on files, Overview of File organizations, and Indexing structures for Files (Primary Index, Secondary Index, Clustering Index)

## Practical / Lab work to be performed

(N.B.: Students need to connect Database using JDBC and need to perform following experiments)

Create and use the following database schemas to answer the given queries.

### EMPLOYEE Schema

FIELD	TYPE	NULL	KEY	DEFAULT
Eno	Char(3)	NO	PRI	NIL
Ename	Varchar(50)	NO		NIL
Job_type	Varchar(50)	NO		NIL
Manager	Char(3)	Yes	FK	NIL

Hire_date	Date	NO		NIL
Dno	Integer	YES	FK	NIL
Commision	Decimal(10,2)	YES		NIL
Salary	Decimal(7,2)	NO		NIL

### DEPARTMENT Schema

FIELD	TYPE	NULL KEY	DEFAULT
Dno	Integer	No PRI	NULL
Dname	Varchar(50)	Yes	NULL
Location	Varchar(50)	Yes	New Delhi

### Query List

(N.B: Student has to perform **any twenty** of the following queries)

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE\_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than \$2850.
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in the range of \$1500 and \$2850.
8. Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Query to display Name and Hire Date of every Employee who was hired in 1981.
10. Query to display Name and Job of all employees who don't have a current Manager.
11. Query to display the Name, Salary and Commission for all the employees who earn commission.
12. Sort the data in descending order of Salary and Commission.
13. Query to display Name of all the employees where the third letter of their name is \_A'.
14. Query to display Name of all employees either have two \_R's or have two \_A's in their name and are either in Dept No = 30 or their Manger's Employee No = 7788.
15. Query to display Name, Salary and Commission for all employees whose Commission Amount is 14 greater than their Salary increased by 5%.
16. Query to display the Current Date.
17. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.
18. Query to display Name and calculate the number of months between today and the date each employee was hired.
19. Query to display the following for each employee <E-Name> earns < Salary> monthly but wants < 3 \* Current Salary >. Label the Column as Dream Salary.

20. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with J, 'A' and M.
21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.
23. Query to display Unique Listing of all Jobs that are in Department # 30.
24. Query to display Name, Dept Name of all employees who have an A in their name.
25. Query to display Name, Job, Department No. And Department Name for all the employees working at the Dallas location.
26. Query to display Name and Employee no. Along with their Manger's Name and the Manager's employee no; along with the Employees 'Name who do not have a anager.
27. Query to display Name, Dept No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns a commission.
28. Query to display Name and Salaries represented by asterisks, where each asterisk (\*) signifies \$100.
29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees
30. Query to display the number of employees performing the same Job type functions.
31. Query to display the no. of managers without listing their names.
32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
33. Query to display Name and Hire Date for all employees in the same dept. as Blake.
34. Query to display the Employee No. And Name for all employees who earn more than the average salary.
35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a T.
36. Query to display the names and salaries of all employees who report to King.
37. Query to display the department no, name and job for all employees in the Sales department.

## Reference Books

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.
3. A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts 6<sup>th</sup> Edition, McGrawHill, 2010.
4. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education,2013.

## **CSC-HC-5016: Internet Technologies**

4 Lectures, 4 Practical, Credits 6 (4+2)

Theory: 60 Lectures      Practical: 60 Lectures

### **UNIT 1: Fundamentals (5 lectures)**

Internet Technologies, WWW, Client Server model, P2P model, Web Server, HTTP, DNS, SMTP, Well known port, Front end fundamentals – HTML, HTML 5, CSS etc.

### **UNIT 2: JavaScript (15 lectures)**

Data types, operators, functions, control structures, Events and Event handling, JavaScript Libraries- string, date and time, Regular expressions etc.

### **UNIT 3: Java (5 lectures)**

Use of Objects, Array and Array List class

### **UNIT 4: 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.

### **UNIT 4: JSP (15 lectures)**

Introduction to JavaServer Pages, 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.

### **UNIT 5: Java Beans (10 lectures)**

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

### **Practical / Lab work to be performed**

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 numbers.
5. A person deposits Rs 1000 in a fixed account yielding 5% interest. Compute the amount in the account at the end of each year for n years. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.

### **Reference 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.



# **CSC-HC-5026: Theory of Computation**

5 Lectures, 1 Tutorial, Credits 6 (5+1)  
Theory: 60 Lectures      Tutorial: 15 Lectures

**UNIT 1: Languages** **(16 Lectures)**  
Alphabets, string, language, Basic Operations on language, Concatenation, Kleene Star

**UNIT 2: Finite Automata and Regular Languages** **(28 Lectures)**  
Regular Expressions, Transition Graphs, Deterministic 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.

**UNIT 3: Context free languages** **(16 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.

## **Reference 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. Hopcroft, 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

# CSC-HC-6016: Artificial Intelligence

4 Lectures, 4 Practical, Credits 6 (4+2)

Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Introduction (6 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.

## **UNIT 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.

## **UNIT 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)

## **UNIT 4: Dealing with Uncertainty and Inconsistencies (8 Lectures)**

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

## **UNIT 5: Understanding Natural Languages (6 Lectures)**

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

## **Practical / Lab work to be performed**

**(N.B: Student has to perform any ten of the following experiments)**

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<sup>th</sup> element into every list at all levels.
6. Write a Prolog program to remove the n<sup>th</sup> item from a list.
7. Write a Prolog program, remove n<sup>th</sup> (Before, After) that asserts the After list is the Before list with the removal of every n<sup>th</sup> 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, Reversed List)` 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.

## Reference Books

1. DAN.W. Patterson, Introduction to A.I and Expert Systems – PHI, 2007.
2. Russell & Norvig, Artificial Intelligence-A Modern Approach, LPE, Pearson Prentice Hall, 2nd 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.

# CSC-HC-6026: Computer Graphics

4 Lectures, 4 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Introduction** **(5 Lectures)**

Basic elements of Computer Graphics, Applications of Computer Graphics

## **UNIT 2: Graphics Hardware** **(8 Lectures)**

Input Devices: Keyboard, Mouse, Trackball & Space ball, Joystick, Data Glove, Digitizers, Image Scanners, Touch panels, Light Pens systems.

Output display devices: Refresh CRT, Raster-Scan display and Random-scan display technique, Color display techniques-Beam penetration method and Shadow-mas kmethod, Direct view storage tubes, Emissive & Non-emissive flat-panel, Displays- Plasma panels, Thin-film electrostatic displays, LED and LCD monitor, Three-dimensional viewing devices and Virtual-Reality systems

Display processor: Raster-scan systems, Random-scan systems

## **UNIT 3: Fundamental Techniques in Graphics** **(22 Lectures)**

Line-drawing algorithms: DDA algorithm and Bresenham's Line drawing Algorithm, Midpoint Algorithm for Circle and Ellipse Generation, Curve generation.

Attributes for output primitives: Area-filling Algorithms - Scan-line Polygon-fill

2-D Geometric Transformations: Basic transformations-translation, Rotation and Scaling Matrix representations and Homogeneous Co-ordinate representations, Composite transformations among translation, Rotation and Scaling

2-D viewing: Definition, Viewing transformation pipeline, Window-to-viewport Co-ordinate transformation.

2-D Clipping : Concept and Algorithm: Point clipping, Line clipping - Cohen-Sutherland algorithm, Area clipping, Text clipping, Polygon clipping.

3-D concepts: Display methods-Parallel projection, perspective projection

3-D geometric transformations: Transformation, Translation, Rotation and Scaling around axes, 3-D Viewing Projections – Parallel and Perspective.

## **UNIT 4: Geometric Modeling** **(10 Lectures)**

Representing curves and surface, Bezier curves and surfaces – Definition of Bezier curve and its properties, Algorithms for Bezier curves and surfaces, Hermite curve

**UNIT 5: Visible Surface determination****(8 Lectures)**

Definition, Algorithms for visible surface detection – Depth buffer method, A-buffer method, Ray-casting method, curved surfaces

**UNIT 6: Surface rendering****(7 Lectures)**

Definition and importance, light sources, Basic illumination models-Ambient light, Diffuse reflection, Specular reflector and Phong model

**Practical / Lab work to be performed**

(N.B: Student has to perform **any five** of the following experiments)

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.

**Reference Books**

1. J.D.Foley, A. Van Dam, van Dam, 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.

# CSC-HE-5016: Microprocessor

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Internal Organization of 8085A microprocessor (6 Lectures)**

User Programmable registers, PC, SP, accumulator, flags, data bus, address bus, control bus, instruction word size, opcode format, data format, memory addressing, I/O addressing, address decoding for memory and I/O.

## **UNIT 2: 8085A microprocessor architecture (12 Lectures)**

Pinout and internal architecture of 8085A microprocessor, multiplexed address/data bus, control and status signal, de-multiplexing of control signals, other signals, bus timings, fetch decode and execute cycle, timing diagram for opcode fetch memory read and memory write, interfacing memory and I/O.

## **UNIT 3: Assembly language programming in 8085A microprocessor (20 Lectures)**

Complete instruction set in detail, programming examples, logic operation, counters and time delays, stack and subroutine, processing arrays, bit manipulation.

## **UNIT 4: Interfacing (12 Lectures)**

In and OUT instruction, decoding addresses, Interfacing LED, relay, seven segment display, switch, keyboard.

## **UNIT 5: Interrupt (10 Lectures)**

Vectored interrupts, interrupt priorities, general purpose programmable peripheral devices, 8255A control and status registers, programming 8255A, introduction to 8279, 8254 and 8237 (block diagrams and basic functions).

## **Practical / Lab work to be performed**

(N.B: Student has to perform **any eight** of the following experiments)

1. Write a program to add two 8 bit numbers & store it in a memory location 8820h.
2. Write a program to copy a block of memory from one location 8820h to another location 8840h.
3. Write a program to perform the subtraction of two 16 bit numbers.
4. Write a program to load two unsigned numbers in register b & c. Subtract b from c. If the result is in 2's complement, convert the result in absolute magnitude & display it.
5. Write a program to find the difference of two numbers & store the result in a memory location 8830h.
6. Write a program to perform  $x \square y \square 5$ , where x and y are 16-bit numbers.
7. Write a program for 4 digit BCD addition and subtraction.(16 bit)
8. Write a program to multiply two 8-bit numbers.
9. Write a program to sort an array.

10. Write a program for binary to ascii conversion
11. Write a program for BCD to binary conversion.
12. Write a program to find seven segment codes for given numbers.

### **Recommended Books**

1. Ramesh S. Gaonkar: Microprocessor Architecture, Programming and Application with the 8085, PENRAM INTL. PUBLISHING (INDIA) PVT. LTD., sixth edition
2. D.A.Godse A.P.Godse: Microprocessor Techniques, Technical publications, Fourth revised edition,2008

## CSC-HE-5026: Numerical Methods

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

### UNIT 1: (8 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

### UNIT 2: (12 Lectures)

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

### UNIT 3: (10 Lectures)

Iterative methods: Jacobi and Gauss-Seidel iterative methods, Interpolation: Lagrange's form and Newton's form

### UNIT 4: (12 Lectures)

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

### UNIT 5: (9 Lectures)

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

### UNIT 6: (9 Lectures)

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 4<sup>th</sup> order Runge-Kutta method, Finite difference method for linear ODE

### Practical / Lab work to be performed

(N.B: Student has to perform **any six** of the following experiments)

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.

## **Reference 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)

## **CSC-HE-5036: Project Work / Dissertation**

Credits 6

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.

# **CSC-HE-6016: Network Programming**

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

**UNIT 1: Transport Layer Protocols:** **(12 Lectures)**  
TCP, UDP, SCTP protocol.

**UNIT 2: Socket Programming:** **(24 Lectures)**  
Socket Introduction, TCP Sockets, TCP Client/Server Example, Signal handling, I/O Multiplexing using Sockets, Socket Options, UDP Sockets, UDP Client Server example, Address Lookup using Sockets.

**UNIT 3: Network Applications:** **(12 Lectures)**  
Remote Logging, Email, WWW and HTTP.

**UNIT 4: LAN administration:** **(12 Lectures)**  
Linux and TCP/IP Networking, Network Management and Debugging.

## **Practical / Lab work to be performed**

Practical exercises based on concepts listed in theory.

## **Reference Books**

1. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Unix Network Programming, The sockets Networking API, Vol. 1, 3rd Edition, PHI.2003
2. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM Publishing Company Ltd.,2003
3. Nemeth Synder & Hein, Linux Administration Handbook, Pearson Education, 2nd Edition,2010
4. R. Stevens, Unix Network Programming, PHI 2nd Edition,1990

# CSC-HE-6026: Introduction to Data Sciences

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Data Scientist's Tool Box (12 Lectures)**

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

## **UNIT 2: R Programming Basics (12 Lectures)**

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

## **UNIT 3: Getting and Cleaning Data (8 Lectures)**

Obtaining data from the web, from APIs, from databases and from colleagues in various formats. basics of data cleaning and making data —tidy.

## **UNIT 4: Exploratory Data Analysis (16 Lectures)**

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

## **UNIT 5: Reproducible Research (12 Lectures)**

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.

## **Practical / Lab work to be performed**

(N.B: Student has to perform **any seven** of the following experiments)

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. Their 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
  - iv. 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.

## Reference Books

1. Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Frontline" 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" 1st 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, LinkedIn, Goole+, GitHub, and More", Second Edition, by O'Reilly Media, 2013.

# CSC-HE-6036: System Programming

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Assemblers & Loaders, Linkers** (12 Lectures)

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

## **UNIT 2: Introduction** (4 Lectures)

Overview of compilation, Phases of a compiler

## **UNIT 3: Lexical Analysis** (8 Lectures)

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

## **UNIT 4: Parsing:** (12 Lectures)

Top Down Parsing- LL parser, Bottom up parsing- LR parser, LALR parser, yacc.

## **UNIT 5: Intermediate representations** (12 Lectures)

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

## **UNIT 6: Storage organization** (6 Lectures)

Activation records, stack allocation

## **Unit 7: Code Generation** (6 Lectures)

Object code generation

## **Practical / Lab work to be performed**

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.

## **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, 3<sup>rd</sup> 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

# CSC-HE-6046: Data Mining

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Overview** **(12 Lectures)**

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

## **UNIT 2: Association Rule Mining** **(15 Lectures)**

What is an association rule? Mining association rules, frequent sets and border sets, algorithms for mining association rules –Apriori algorithm, Pincer-Search algorithm, DIC (Dynamic Item set Counting) algorithm, Border algorithm, Concepts of Maximal frequent sets and closed frequent sets

## **UNIT 3: Clustering** **(18 Lectures)**

Concept of Similarity measure and Distance Measure, Euclidean distance, Manhattan distance, Cosine similarity, Jacquard coefficient, Different types of data in clustering, Partitional vs Hierarchical Clustering, Partitional clustering methods –k-means, k-medoids, PAM, CLARA, CLARANS, Hierarchical clustering methods –BIRCH, CURE, Density based clustering methods-DBSCAN

## **UNIT 4: Classification and regression technique** **(15 Lectures)**

Introduction, Clustering versus Classification, decision tree construction principle, decision tree generation algorithms –CART, ID3

The Problem of over fitting and techniques for reducing it, Scalability and Data management issues in data mining algorithms, measures of interestingness.

## **Practical / Lab work to be performed**

(N.B: Student has to perform **any five** of the following experiments)

1. Write a computer program to implement A priori algorithm using two different data structures for storing the item sets in the level wise generation of candidate sets.
2. Write a computer program to implement A priori algorithm using hashing to store the item sets in the level wise generation of candidate sets.
3. Write a computer program to implement the Pincer Search algorithm.
4. Write a computer program to implement the DIC (Dynamic Item set) algorithm.
5. Write a computer program to implement the Promoted Border Algorithm.
6. Write a computer program to implement the k-means algorithm and test it using the different similarity measures stated in the syllabus.
7. Write a computer program to implement the PAM algorithm.
8. Write a computer program to implement the CLARA algorithm.
9. Write a computer program to implement the CLARANS algorithm.
10. Write a computer program to implement the DBSCAN algorithm.

11. Write a computer program to implement the BIRCH algorithm.
12. Write a computer program to implement the CART algorithm.
13. Write a computer program to implement the ID3 algorithm.

## **Reference Books**

1. A.K. Puzari, Data Mining Techniques, University Press.
2. J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufman. 2001.
3. P. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Pearson Education (LPE), 2009.
4. G.K. Gupta, Introduction to Data Mining with Case Studies, PHI, 2006.



# CSC-SE-3014: UNIX/LINUX Programming

02 Lectures, 04 Practical, Credits 4 (2+2)  
Theory: 20 Lectures      Practical: 20 Lectures

## **UNIT 1: Introduction (6 Lectures)**

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, System processes (an overview), External and internal commands, Creation of partitions in OS, Processes and its creation phases – Fork, Exec, wait

## **UNIT 2: User Management and the File System (6 Lectures)**

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

## **UNIT 3: Shell introduction and Shell Scripting (8 Lectures)**

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)

## **Practical / Lab work to be performed Based on Linux** (N.B: Student has to perform **any ten** of the following experiments)

1. Write a shell script to check if the number entered at the command line is prime or not.
2. Write a shell script to modify `—call` command to display calendars of the specified months.
3. Write a shell script to modify `—call` command to display calendars of the specified range of months.
4. Write a shell script to accept a login name. If not a valid login name display message –  
Entered login name is invalidl.
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 `—whol` 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.
20. Program to show the pyramid of special character —\*|.

## 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

# CSC-SE-3024: Programming in Python

02 Lectures, 04 Practical, Credits 4 (2+2)  
Theory: 20 Lectures      Practical: 20 Lectures

## **UNIT 1: Planning the Computer Program (2 Lecture)**

Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation

## **UNIT 2: Techniques of Problem Solving (2 Lecture)**

Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming

## **UNIT 3: Overview of Programming (4 Lecture)**

Structure of a Python Program, Elements of Python

## **UNIT 4: Introduction to Python (4 Lecture)**

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

## **UNIT 5: Creating Python Programs (4 Lecture)**

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

## **UNIT 6: Python File Operations (4 Lecture)**

Understanding read functions, read(), readline() and readlines(), Understanding write functions, write() and writelines(), Programming using file Operations, Reading config files in python, Writing log files in python.

## **Practical / Lab work to be performed**

### **Section: A ( Simple programs)**

**(N.B:** Student has to perform **any four** of the following experiments)

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

Grade A: Percentage  $\geq 80$

Grade B: Percentage  $\geq 70$  and  $< 80$

Grade C: Percentage  $\geq 60$  and  $< 70$

Grade D: Percentage  $\geq 40$  and  $< 60$

Grade E: Percentage<40

3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. Write a program to display the first n terms of Fibonacci series.
5. Write a program to find factorial of the given number.
6. Write a program to find sum of the following series for n terms:  
$$1 - 2/2! + 3/3! - \dots - n/n!$$
7. Write a program to calculate the sum and product of two compatible matrices.

**Section: B (Visual Python)**

(N.B: Student has to perform **any four** of the following experiments)

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

1. Write a menu-driven program to create mathematical 3D objects  
(i) curv, (ii) sphere (iii) cone (iv) arrow (v) ring (vi) cylinder.
2. Write a program to read n integers and display them as a histogram.
3. Write a program to display sine, cosine, polynomial and exponential curves.
4. Write a program 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. Write a program 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  $t \geq 0$ .
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. Write a program to show a ball bouncing between 2 walls. (Optional)

**Reference Books**

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. Python Tutorial/Documentation [www.python.org](http://www.python.org) 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/>

# CSC-SE-3034: HTML Programming

02 Lectures, 04 Practical, Credits 4 (2+2)  
Theory: 20 Lectures      Practical: 20 Lectures

## **UNIT 1: The Basics** **(3 Lectures)**

Introduction to HTML, the Head, the Body, Colors, Attributes, Check box, Radio Button, Text, TextArea, Lists, ordered and unordered

## **UNIT 2: HTML Formatting** **(3 Lectures)**

New Paragraph, Line Break, Blank Space, Preformatted text, Div element Bold text, Important text, Italic text, Emphasized text, Marked text, Small text, Deleted text, Inserted text, Subscript text, Superscript text, HTML quotations, HTML Comments, HTML colors

## **UNIT 3: Links** **(3 Lectures)**

Introduction, Relative Links, Absolute Links, Link Attributes, Using the ID Attribute to Link within a Document

## **UNIT 4: Images** **(2 Lectures)**

Putting an Image on a Page, Using Images as Links, Putting an Image in the Background

## **UNIT 5: Tables** **(5 Lectures)**

Creating a Table, Table Headers, Captions, Spanning Multiple Columns, Styling Table

## **UNIT 6: Forms** **(4 Lectures)**

Basic Input and Attributes, Other Kinds of Inputs, Styling forms with CSS.

## **Practical / Lab work to be performed**

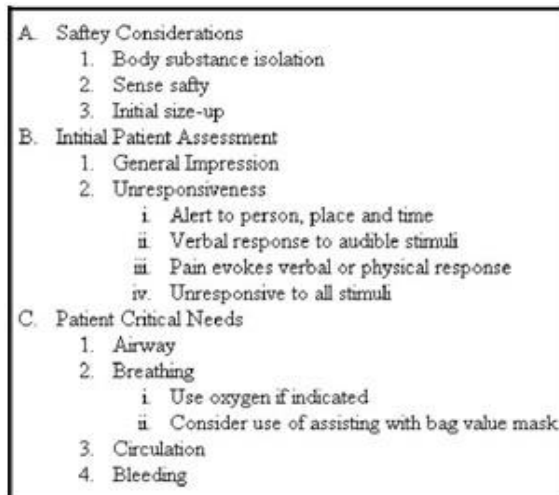
1. Create an HTML document with the following formatting options:

- (i) Bold
- (ii) Italics
- (iii) Underline
- (iv) Headings (Using H1 to H6 heading styles)
- (v) Font (Type, Size and Color)
- (vi) Background (Colored background/Image in background)
- (vii) Paragraph
- (viii) Line Break
- (ix) Horizontal Rule
- (x) Pre tag

2. Create an HTML document which consists of:

- (i) Ordered List

- (ii) Unordered List
- (iii) Nested List
- (iv) Image



- 3. Create an HTML document which implements Internal linking as well as External Linking.
- 4. Create a table using HTML which consists of columns for Roll No., Student's name and grade.

Result		
Roll No	Name	Grade

- 5. Create a Table with the following view:

				Place an Image Here

- 6. Create a form using HTML which has the following types of controls:
  - (i) Text Box

- (ii) Option/radio buttons
- (iii) Check boxes
- (iv) Reset and Submit buttons

**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:

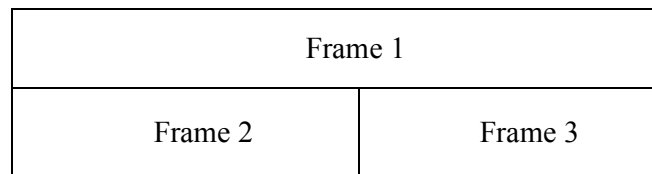
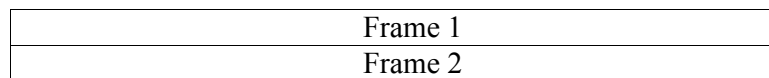
How did you hear about XYZ News Magazine and Emails?

Here on the Web  In a magazine  Television  Other

Would you like to be on our regular mailing list?

Yes, we love junk emails

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



## Reference Books

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

# CSC-SE-4014: Android Programming

02 Lectures, 04 Practical, Credits 4 (2+2)  
Theory: 20 Lectures      Practical: 20 Lectures

## **UNIT 1: Introduction** **(2 Lectures)**

What is Android and its importance, Brief history about Android, Pre-requisites to learn Android, Development tools, Android architecture-software stack.

## **UNIT 2: Get started with Android** **(4 Lectures)**

Installation of Android Studio, SDK, emulator, Creating your first Project App, Explore project structure- Manifest, java, res, gradle, Activity, Layout, Common UI components, Creating UI through code, Run the application in Emulator and Android device.

## **UNIT 3: Activities** **(3 Lectures)**

Creating a new activity, Application context, Intent, Activity Life cycle, Communicating among Activities, connecting to internet resource, WebView.

## **UNIT 4: Designing User Interface** **(3 Lectures)**

Using Button control, ArrayList, Spinner, ListView, GridView, Adapters, Sidebar Menus, Notification (toast, Status bar), Adding image, audio, Video.

## **UNIT 5: Background Task and Local File Storage** **(3 Lectures)**

Broadcast Receiver, services in Android, Reading/writing local data, Accessing the Internal File system.

## **UNIT 6: Database** **(5 Lectures)**

SQLite- Introducing SQLite, SQLiteOpenHelper and creating a database, Opening and closing a database, Inserts, updates, and deletes operation, Data Binding, using content provider, firebase, storing and sharing data in the cloud.

## **Practical / Lab work to be performed**

**(N.B:** Student has to perform **any six** of the following experiments)

1. Create an application with two fields (Number 1, Number2). On clicking button the sum of the two numbers are displayed on another text field.
2. Create a Login application with two fields (username and password). On successful login go to next screen, also pass user name to next screen. And on failing login, alert user using Toast.
3. Create an application with two fields (name and value). On clicking button, the value on the two fields passes to the next screen and display it on text field.
4. Create an application with a button. On clicking the button open browser with an URL.(open google page)
5. Create a spinner with names of some countries. On selecting a particular country, Display a toast message with currency name of that country.



6. Create an application with Navigation side bar menu item. On clicking the sidebar menu item, it open activity page of that particular menu item.
7. Create an application with static Broadcast Receiver features. When a SMS comes, it displays an alert message.
8. Create an application to display an image and button. On button click the image is changed. The images are put inside the resource folder.
9. Create an application with a button. On clicking it, notification is displayed on top of the app. When swipe and click it display second activity page.
10. Create an application with two button for start and stop services. On clicking start button it will play default ringtone in the background. On clicking stop button it stops.
11. Create an application to play a video.
12. Create an application with a text field. On clicking the Add button the content of the text field is save to a file in internal memory storage. On clicking the Load button the content of the file in the internal storage is displayed again in the text field.
13. Create an application to create a database in Sqlite with a table of user information (id, name, marks etc.). The application can also perform insert, retrieve, update and delete operation.

## **Reference Books**

1. Grant Allen, Beginning Android 4, A press, 2012.
2. Wei-Meng Lee, Beginning android 4 application Development, John Wiley & sons, Inc, 2012.
3. Android programming for Beginners by John Horto, Packt Publishing Ltd.

# CSC-SE-4024: PHP Programming

02 Lectures, 04 Practical, Credits 4 (2+2)  
Theory: 20 Lectures      Practical: 20 Lectures

## **UNIT 1: Introduction to PHP** **(4 Lectures)**

PHP introduction, inventions and versions, important tools and software requirements (like Web Server, Database, Editors etc.), PHP with other technologies, scope of PHP, Basic Syntax, PHP variables and constants, Types of data in PHP , Expressions, scopes of a variable (local, global), PHP Operators : Arithmetic, Assignment, Relational , Logical operators, Bitwise , ternary and MOD operator, PHP operator Precedence and associativity

## **UNIT 2: Handling HTML form with PHP** **(2 Lectures)**

Capturing Form Data, GET and POST form methods, Dealing with multi value fields, Redirecting a form after submission

## **UNIT 3: PHP conditional events and Loops** **(3 Lectures)**

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

## **UNIT 4: PHP Functions** **(3 Lectures)**

Function, Need of Function, declaration and calling of a function, PHP Function with arguments, Default Arguments in Function, Function argument with call by value, call by reference, Scope of Function Global and Local

## **UNIT 5: String Manipulation and Regular Expression** **(4 Lectures)**

Creating and accessing String , Searching & Replacing String, Formatting, joining and splitting String , String Related Library functions, Use and advantage of regular expression over inbuilt function, Use of preg\_match(), preg\_replace(), preg\_split() functions in regular expression

## **UNIT 6: Array** **(4 Lectures)**

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

## **Practical / Lab work to be performed**

(N.B: Student has to perform **any ten** of the following experiments)

1. Create a PHP page using functions for comparing three integers and print the largest number.
2. Write a function to calculate the factorial of a number (non-negative integer). The function accept the number as an argument.
3. Write a program to check whether the given number is prime or not.

4. Create a PHP page which accepts string from user. After submission that page displays the reverse of provided string.
5. Write a PHP function that checks if a string is all lower case.
6. Write a PHP script that checks whether a passed string is palindrome or not? ( A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run)
7. Write a program 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 —Hellol 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.

## 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.

## **CSC-SE-4034: R Programming**

02 Lectures, 04 Practical, Credits 4 (2+2)  
Theory: 20 Lectures      Practical: 20 Lectures

### **UNIT 1: Introduction** **(6 Lectures)**

Overview and History of R, Getting Help, R Data Types and Objects, Subsetting, Vectorized, Operations, Reading and Writing Data.

### **UNIT 2: Control** **(8 Lectures)**

Control Structures, Functions, lapply, tapply, split, mapply, apply, Scoping Rules, Coding Standards.

### **UNIT 3: Simulation** **(6 Lectures)**

Loop functions, Debugging Tools, Simulation, R Profiler.

### **Practical / Lab work to be performed**

**(N.B:** Student has to perform **any eight** of the following experiments)

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 or not.
7. Implement the sorting algorithms: Selection sort, Insertion sort, Bubble Sort
8. Implement linear search and binary search algorithm.
10. Implement matrices addition, subtraction and multiplication

### **Reference Books**

1. William N. Venables and David M. Smith, An Introduction to R. 2<sup>nd</sup> Edition. Network Theory Limited.2009
2. Norman Matloff, The Art of R Programming - A Tour of Statistical Software Design, No Starch Press.2011

## GENERAL ELECTIVES

### CSC-HG-1016: Problem Solving using Computer (CSC-RC-1016)

04 Lectures, 04 Practical, Credits 6 (4+2)

Theory: 60 Lectures      Practical: 60 Lectures

End Semester Marks:

Theory: 60 Marks, Practical: 20 Marks

Internal Marks:

Sessional: 10 Marks, Practical: 6 Marks, Attendance: 4 Marks

#### **UNIT 1: Computer Fundamentals (3 Lectures)**

Introduction to Computers, Characteristics of Computers, Uses of computers, Types and generations of Computers

#### **UNIT 2: Basic Computer Organization (4 Lectures)**

Units of a computer, CPU, ALU, memory hierarchy, registers, I/O devices

#### **UNIT 3: Planning the Computer Program (3 Lectures)**

Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation

#### **UNIT 4: Techniques of Problem Solving (4 Lectures)**

Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming

#### **UNIT 5: Overview of Programming (4 Lectures)**

Structure of a Python Program, Elements of Python

#### **UNIT 6: Introduction to Python (8 Lectures)**

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

#### **UNIT 7: Creating Python Programs (10 Lectures)**

Input and Output Statements, Control statements (Looping- while Loop, for Loop, Loop Control, Conditional Statement- if...else, Difference between break, continue and pass)

#### **UNIT 8: Structures (10 Lectures)**

Numbers, Strings, Lists, Tuples, Dictionary, Date & Time, Modules, Defining Functions, Exit function, default arguments

#### **UNIT 9: Introduction to Advanced Python (14 Lectures)**

## Practical / Lab work to be performed (using 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 user's choice.
2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria

Grade A: Percentage  $\geq 80$

Grade B: Percentage  $\geq 70$  and  $< 80$

Grade C: Percentage  $\geq 60$  and  $< 70$

Grade D: Percentage  $\geq 40$  and  $< 60$

Grade E: Percentage  $< 40$

3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. WAP to display the first n terms of Fibonacci series.
5. WAP to find factorial of the given number.
6. WAP to find sum of the following series for n terms:  $1 - 2/2! + 3/3! - \dots - n/n!$
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  $t \geq 0$ .

6. A population of 1000 bacteria is introduced into a nutrient medium. The population  $p$  grows as follows:

$$P(t) = (15000(1+t))/(15 + e^t)$$

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 ( $s = (v*v - u*u) / 2*a$ )

## Reference Books

1. P. K. Sinha & Priti Sinha, "Computer Fundamentals", BPB Publications, 2007.
2. Dr. Anita Goel, Computer Fundamentals, Pearson Education, 2010.
3. T. Budd, Exploring Python, TMH, 1st Ed, 2011
4. Python Tutorial/Documentation [www.python.org](http://www.python.org) 2010
5. Allen Downey, Jeffrey Elkner, Chris Meyers, How to think like a computer scientist : learning with Python, Freely available online.2012
6. <http://docs.python.org/3/tutorial/index.html>
7. <http://interactivepython.org/courselib/static/pythononds>
8. <http://www.ibiblio.org/g2swap/byteofpython/read/>

## **CSC-HG-1026: Multimedia and Applications**

04 Lectures, 04 Practical, Credits 6 (4+2)

Theory: 60 Lectures      Practical: 60 Lectures

End Semester Marks:

Theory: 60 Marks, Practical: 20 Marks

Internal Marks:

Sessional: 10 Marks, Practical: 6 Marks, Attendance: 4 Marks

### **UNIT 1: Multimedia (6 Lectures)**

Introduction to multimedia, components, uses of multimedia, multimedia applications, virtual reality

### **UNIT 2: Text (4 Lectures)**

Fonts & Faces, Using Text in Multimedia, Font Editing & Design Tools, Hypermedia & Hypertext

### **UNIT 3: Images (6 Lectures)**

Still Images – bitmaps, vector drawing, 3D drawing & rendering, natural light & colors, computerized colors, color palettes, image file formats.

### **UNIT 4: Sound (6 Lectures)**

Digital Audio, MIDI Audio, MIDI vs Digital Audio, Audio File Formats

### **UNIT 5: Video (8 Lectures)**

How video works, analog video, digital video, video file formats, video shooting and editing.

### **UNIT 6: Animation (10 Lectures)**

Principle of animations, animation techniques, animation file formats.

### **UNIT 7: Internet and Multimedia (6 Lectures)**

www and HTML, multimedia on the web – web servers, web browsers, web page makers and site builders.

### **UNIT 8: Making Multimedia (14 Lectures)**

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

### **Practical / Lab work to be performed**

(N.B: Student has to perform **any seven** of the following experiments)



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

Implement the followings using Flash-

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

## References Books

1. Tay Vaughan, —Multimedia: Making it work, TMH, Eighth edition.2011
2. Ralf Steinmetz and KlaraNaharstedt, —Multimedia: Computing, Communications Applications, Pearson.2012
3. Keyes, —Multimedia Handbook, TMH,2000.
4. K. Andleigh and K. Thakkar, —Multimedia System Design, PHI.2013

## **CSC-HG-1036: Fundamentals of Computer Applications (TCA-RC-1016)**

04 Lectures, 04 Practical, Credits 6 (4+2)

Theory: 60 Lectures      Practical: 60 Lectures

End Semester Marks:

Theory: 60 Marks, Practical: 20 Marks

Internal Marks:

Sessional: 10 Marks, Practical: 6 Marks, Attendance: 4 Marks

### **UNIT 1: Major components of a computer (12 Lectures)**

Block diagram, A brief introduction of CPU, Main Memory, I/O units, RAM, ROM, Keyboard, display, mouse, printers etc, Hardware, Software, Secondary storage devices (hard disks, optical disks, flash memory), backup devices, Bootstrapping, Representation of Information, Number Systems-binary, octal, decimal and hexadecimal, ASCII, EBDIC, BCD Gray code, Unicode. Conversion of bases

### **UNIT 2: Algorithms (10 Lectures)**

Concept of algorithm and flow chart, Writing simple algorithms and drawing flow charts for simple problems like finding sum, max, min, average of a list of numbers etc.

### **UNIT 3: Operating System (10 Lectures)**

Evolution of OS, types of OS, functions of OS, Case study on any OS. [No questions on theory paper will be set on these topics] System features, Software features, File structure, File handling, Installation of OS, Hardware & Software requirements. Computer Networks

### **UNIT 4: Data communication (10 Lectures)**

Transmission Media- Coaxial, UTP, Optical-Fiber, Wireless, Components of Computer Networks, Types of wireless communication (mobile, WiFi, WiMAX, Bluetooth, Infrared – concept and definition only)

### **UNIT 5: Internet (10 Lectures)**

Evolution of internet, Basic internet terms( Client, Server, MODEM, Web page, Web site, Home page, Browser, URL, ISP, Web server, Download & Upload, Online & Offline), Internet applications (e-mail, search engines, ftp, VOIP, Video Conferencing, Audio-Video streaming, Chatting).

### **UNIT 6: Computer Security (8 Lectures)**

Definition, Viruses and Worms, Antivirus, Digital Signature, Software Piracy, Firewall. Hacking and Cracking (basic concepts only for these topics will have to be given).

## **Practical / Lab work to be performed**

**Introduction to Open Office/MS Office/Libre Office**

**(N.B:** Students have to perform the following experiments and are encouraged to work in the Linux platform)

1. Create a new folder and do the following:
  - Make a word processing document in it.
  - Make a Spreadsheet document in it.
  - Make a new folder in it
  - Rename the initial folder
  - Move the initial folder
  - Copy the initial folder.
  - Delete the initial folder
2. Implement the various well known features of the operating system such as Painting, System tools, Entertainment tools etc.
3. Implement various display properties by right clicking on the Desktop.
4. Explore the taskbar
5. Set the wall paper and screen saver.
6. Set the data/time.

### **Word Processing Tool**

1. Create a document and
  - a. Put Bullets and Numbers
  - b. Apply various Font parameters.
  - c. Apply Left, Right, and Centre alignments.
  - d. Apply hyperlinks
  - e. Insert pictures
  - f. Insert ClipArt
  - g. Show the use of WordArt
  - h. Add Borders and Shading
  - i. Show the use of Find and Replace.
  - j. Apply header/footers
2. Create any document and show the use of File Versions.
3. Create any document and show the difference between paste and paste special.
4. Create a document to show the use of Washout/Watermark.
5. Implement the concept of mail merge.
6. Implement the concept of macros.
7. Implement the concept of importing a file/document.
8. Implement the concept of merging the documents.
9. Create a student table and do the following:
  - a. Insert new row and fill data
  - b. Delete any existing row
  - c. Resize rows and columns
  - d. Apply border and shading
  - e. Apply merging/splitting of cells
  - f. Apply sort
  - g. Apply various arithmetic and logical formulas.
10. Create your resume using General Templates.

**Spreadsheet Tool**

1. Create a student worksheet containing roll numbers, names and total marks. Open a document in Word and insert the excel worksheet using:-

- i) Copy/Paste
- ii) Embedding
- iii) Linking

2. The term wise marks for APS class of 20 students are stored in 3 separate sheets named term1, term2 and term3. Create 4<sup>th</sup> worksheet that contains student names and their total and average marks for the entire year. Give proper headings using headers. Make the column headings bold and italic. The 4<sup>th</sup> worksheet should contain college name as the first line. Make it bold, italic and center it.

3. Using a simple pendulum, plot 1-T and 1-T<sup>2</sup> graph.

I	t1	t2	t3	Mean(t)	T=t/20	T2
70						
80						
90						
100						

4. Consider the following employee worksheet:-

Full Name (First Last)	Grade 1/2/3	Basic Salary	HRA	PF	Gross	Net	(VA) Vehicle Allowance

HRA is calculated as follows:

Grade	HRA %(of Basic)
1	40%
2	35%
3	30%

Gross = Basic + HRA + VA Net = Gross –PF

PF is 8% for all Grades

VA is 15000, 10000 and 7000 for Grades 1, 2 and 3.

- i) Find max, min and average salary of employees in respective Grade
- ii) Count no. of people where VA>HRA
- iii) Find out most frequently occurring grade.
- iv) Extract records where employee name starts with “A” has HRA>10000
- v) Print Grade wise report of all employees with subtotals of net salary and also grand totals. Use subtotal command.

- vi) Extract records where Grade is 1 or 2 and salary is between 10000 and 20000 both inclusive.
5. In a meeting of a marketing department of an organization it has been decided that price of selling an item is fixed at Rs40. It was resolved to increase the sell of more of more items and getting the profit of Rs40,000/. Use Goal Seek of find out how many items you will have to sell to meet your profit figure.
6. To study the variation in volume with pressure for a sample of an air at constant temperature by plotting a graph for P – V and P-I/V. Sample observations are:-

Pressure(P)	Volume (V)	I/V	PV	P/V
75	20			
78.9	19			
83.3	18			
88.2	17			

7. Plot the chart for marks obtained by the students (out of 5) vs. frequency (total number of students in class is 50).
8. Create the following worksheet(s) containing an year wise sale figure of five salesmen in Rs.

Salesman	2002	2003	2004	2005
MOHAN	10000	12000	20000	50000
MITRA	15000	18000	50000	60000
SHIKHA	20000	22000	70000	70000
ROHIT	30000	30000	100000	80000
MANGLA	40000	45000	125000	90000

Apply the following Mathematical & Statistical functions:

- i) Calculate the commission for each salesman under the condition :-
  - a) If total sales is greater than Rs. 3, 00,000/-, then commission is 10% of total sale made by the salesman.
  - b) Otherwise, 4% of total sale.
- ii) Calculate the maximum sale made by each salesman.
- iii) Calculate the maximum sale made in each year.
- iv) Calculate the minimum sale made by each salesman.
- v) Calculate the minimum sale made in each year.
- vi) Count the no. of sales persons.
- vii) Calculate the cube of sales made by Mohan in the year 2002.
- viii) Find the difference in sales by salesman Mitra between the year 2002 and 2003. Find the absolute value of difference.
- ix) Also calculate the Mode, Stddev, Variance, Median for the sale made by each salesman.
- ix) Calculate the year wise Correlation coefficient between the sales man Mohan and Mitra

year wise

9. The following table gives a 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

- i) Calculate total sale year wise.
- ii) Calculate the net sales made by each salesman
- iii) Calculate the commission for each salesman under the condition :-
  - a) If total sales is greater than Rs. 4, 00,000/-, then commission is 5% of total sale made by the salesman.
  - b) Otherwise, 2% of total sale.
- iv) Calculate the maximum sale made by each salesman.
- v) Calculate the maximum sale made in each year.
- vi) Draw a bar graph representing the sale made by each salesman.
- vii) Draw a pie graph representing the sale made by salesmen in year 2001.

10. Consider the following worksheet for APS 1<sup>st</sup> year students:-

S.No.	Name	PH	CH	BY	MT	CS	Total Marks	%	Grade
1									
2									

Grade is calculated as follows:-

- If %  $\geq 90$  Grade A
- If %  $\geq 80$  &  $< 90$  Grade B
- If %  $\geq 70$  &  $< 80$  Grade C
- If %  $\geq 60$  &  $< 70$  Grade D

Otherwise students will be declared fail.

- i) Calculate Grade using if function
- ii) Sort the data according to total marks
- iii) Apply filter to display the marks of the students having more than 65% marks.
- iv) Draw a pie chart showing % marks scored in each subject by the topper of the class.
- v) Draw the doughnut chart of the data as in (iv)
- vi) Enter the S.No. of a student and find out the Grade of the student using VLOOKUP.
- vii) Extract all records where name
  - a) Begins with "A"
  - b) Contains "A"

- c) Ends with “A”

### **Presentation Tool**

1. Make a presentation of College Education System using
  - a. Blank Presentation
  - b. From Design Template
  - c. From Auto Content Wizard
2. Make a presentation on “Wild Life” and apply the following:
  - a. Add audio and video effects
  - b. Apply various Color Schemes
  - c. Apply various animation schemes.
  - d. Apply Slide Show

### **Reference Books**

1. Sushila Madan , Introduction to Essential tools,JBA,2009.
2. Anita Goel, Computer Fundamentals, Pearson, 2012
3. Introduction to Information Technology, ITL solutions Ltd, Pearson Education
4. Fundamentals of Computer by V. Rajaraman.
5. P. K. Sinha & Priti Sinha , “Computer Fundamentals”, BPB Publications, 2007.
6. Dr. Anita Goel, Computer Fundamentals, Pearson Education, 2010.
7. T. Budd, Exploring Python, TMH, 1st Ed, 2011
8. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python , Freely available online.2012

## **CSC-HG-1046: Fundamentals of Information Technology (TIT-RC-1016)**

04 Lectures, 04 Practical, Credits 6 (4+2)

Theory: 60 Lectures      Practical: 60 Lectures

End Semester Marks:

Theory: 60 Marks, Practical: 20 Marks

Internal Marks:

Sessional: 10 Marks, Practical: 6 Marks, Attendance: 4 Marks

### **UNIT 1: Fundamentals (10 Lectures)**

Major components of a computer (A brief introduction of CPU, Main Memory, I/O units), Keyboard, display, mouse, printers etc, Secondary storage devices (hard disks, optical disks, flash memory), backup devices .Computer Software, Bootstrapping

### **UNIT 2: Representation of information (8 Lectures)**

Number system-Binary, Octal, Decimal & Hexadecimal numbers, Characters, ASCII, EBCDIC, BCD, Gray code, Unicode, Conversion of bases

### **UNIT 3: Information Technology Basics (4 Lectures)**

Introduction, Information, Technology, Information Technology (IT), Role in present scenario

### **UNIT 4: Algorithms (10 Lectures)**

Concept of algorithm and flow chart, Writing simple algorithms and drawing flow charts for simple problems like finding sum, max, min, average of a list of numbers etc.

### **UNIT 5: Operating System (8 Lectures)**

Evolution of OS, types of OS, functions of OS, DOS- Internal & external commands.

### **UNIT 6: Computer Networks (8 Lectures)**

Data communication, Transmission Media- Coaxial, UTP, Optical-Fiber, Wireless, Components of Computer Networks, Types of wireless communication (mobile, WiFi, WiMAX, Bluetooth, Infrared –concept and def only)

### **UNIT 7: Internet (8 Lectures)**

Internet-Evolution of Internet, Basic Internet terms( Client, Server, MODEM, Web page, Web site, Homepage, Browser, URL, ISP, Web server, Download & Upload, Online & offline), Internet applications(e-mail, search engines, ftp, VOIP, Video Conferencing, Audio-Video streaming, Chatting).Computer Security, Definition, Viruses and Worms, Antivirus, Digital Signature, Software Piracy, Firewall. Hacking and Cracking (basic concepts only for these topics)



Railways, Airlines, Banking, Inventory Control, Education and Information Kiosks

**Practical / Lab work to be performed**

**(N.B:** Students have to perform the following experiments and are encouraged to work in the Linux platform)

1. Create a new folder and do the following:
  - Make a word processing document in it.
  - Make a Spreadsheet document in it.
  - Make a new folder in it
  - Rename the initial folder
  - Move the initial folder
  - Copy the initial folder.
  - Delete the initial folder
2. Implement the various well known features of the operating system such as Painting, System tools, Entertainment tools etc.
3. Implement various display properties by right clicking on the Desktop.
4. Explore the taskbar
5. Set the wall paper and screen saver.
6. Set the date/time.

**Word Processing Tool**

1. Create a document and
  - a. Put Bullets and Numbers
  - b. Apply various Font parameters.
  - c. Apply Left, Right, and Centre alignments.
  - d. Apply hyperlinks
  - e. Insert pictures
  - f. Insert ClipArt
  - g. Show the use of WordArt
  - h. Add Borders and Shading
  - i. Show the use of Find and Replace.
  - j. Apply header/footers
2. Create any document and show the use of File versions.
3. Create any document and show the difference between paste and paste special.
4. Create a document to show the use of Washout/Watermark.
5. Implement the concept of mail merge.
6. Implement the concept of macros.
7. Implement the concept of importing a file/document.
8. Implement the concept of merging the documents.
9. Create a student table and do the following:
  - a. Insert new row and fill data

- b. Delete any existing row
  - c. Resize rows and columns
  - d. Apply border and shading
  - e. Apply merging/splitting of cells
  - f. Apply sort
  - g. Apply various arithmetic and logical formulas.
10. Create your resume using General Templates.

### **Spreadsheet Tool**

1. Create a student worksheet containing roll numbers, names and total marks. Open a document in Word and insert the excel worksheet using:-
  - i) Copy/Paste
  - ii) Embedding
  - iii) Linking
2. The term wise marks for APS class of 20 students are stored in 3 separate sheets named term1, term2 and term3. Create 4<sup>th</sup> worksheet that contains student names and their total and average marks for the entire year. Give proper headings using headers. Make the column headings bold and italic. The 4<sup>th</sup> worksheet should contain college name as the first line. Make it bold, italic and centerit.
3. Using a simple pendulum, plot 1-T and 1-T<sup>2</sup> graph.

I	t1	t2	t3	Mean(t)	T=t/20	T <sup>2</sup>
70						
80						
90						
100						

4. Consider the following employee worksheet:-

Full Name (First Last)	Grade 1/2/3	Basic Salary	HRA	PF	Gross	Net	(VA) Vehicle Allowance

HRA is calculated as follows:

Grade	HRA %(of Basic)
1	40%
2	35%
3	30%

Gross = Basic + HRA + VA Net = Gross –PF

PF is 8% for all Grades

VA is 15000, 10000 and 7000 for Grades 1, 2 and 3.

- i) Find max, min and average salary of employees in respective Grade
  - ii) Count no. of people where VA>HRA
  - iii) Find out most frequently occurring grade.
  - iv) Extract records where employee name starts with "A" has HRA>10000
  - v) Print Grade wise report of all employees with subtotals of net salary and also grand totals. Use subtotal command.
  - vi) Extract records where Grade is 1 or 2 and salary is between 10000 and 20000 both inclusive.
5. In a meeting of a marketing department of an organization it has been decided that price of selling an item is fixed at Rs40. It was resolved to increase the sell of more of more items and getting the profit of Rs40,000/. Use Goal Seek of find out how many items you will have to sell to meet your profit figure.
6. To study the variation in volume with pressure for a sample of an air at constant temperature by plotting a graph for P – V and P-I/V. Sample observations are:-

Pressure(P)	Volume (V)	I/V	PV	P/V
75	20			
78.9	19			
83.3	18			
88.2	17			

7. Plot the chart for marks obtained by the students (out of 5) vs. frequency (total number of students in class is 50).
8. Create the following worksheet(s) containing an year wise sale figure of five salesmen in Rs.

Salesman	2002	2003	2004	2005
MOHAN	10000	12000	20000	50000
MITRA	15000	18000	50000	60000
SHIKHA	20000	22000	70000	70000
ROHIT	30000	30000	100000	80000
MANGLA	40000	45000	125000	90000

Apply the following Mathematical & Statistical functions:

- i) Calculate the commission for each salesman under the condition:-
  - a) If total sales is greater than Rs. 3, 00,000/-, then commission is 10% of total sale made by the salesman.
  - b) Otherwise, 4% of total sale.
- ii) Calculate the maximum sale made by each salesman.
- iii) Calculate the maximum sale made in each year.
- iv) Calculate the minimum sale made by each salesman.
- v) Calculate the minimum sale made in each year.
- vi) Count the no. of salespersons.
- vii) Calculate the cube of sales made by Mohan in the year2002.

- viii) Find the difference in sales by salesman Mitra between the year 2002 and 2003. Find the absolute value of difference.
  - ix) Also calculate the Mode, Stddev, Variance, Median for the sale made by each salesman.
  - x) Calculate the year wise Correlation coefficient between the sales man Mohan and Mitra year wise
9. The following table gives a year wise sale figure of five salesmen inRs.

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

- i) Calculate total sale yearwise.
  - ii) Calculate the net sales made by eachsalesman
  - iii) Calculate the commission for each salesman under the condition:-
    - a) If total sales is greater than Rs. 4, 00,000/-, then commission is 5% of total sale made by thesalesman.
    - b) Otherwise, 2% of totalsale.
  - iv) Calculate the maximum sale made by eachsalesman.
  - v) Calculate the maximum sale made in eachyear.
  - vi) Draw a bar graph representing the sale made by eachsalesman.
  - vii) Draw a pie graph representing the sale made by salesmen in year2001.
10. Consider the following worksheet for APS 1<sup>st</sup> year students:-

S.No.	Name	PH	CH	BY	MT	CS	Total	%	Grade
							Marks		
1									
2									

Grade is calculated as follows:-

If %> =90 Grade

A

If % >=80 & <90 Grade B

If % >=70 & <80 Grade C

If % >=60 & <70 Grade D

Otherwise students will be declared fail.

- i) Calculate Grade using if function
- ii) Sort the data according to totalmarks
- iii) Apply filter to display the marks of the students having more than 65%marks.

- iv) Draw a pie chart showing % marks scored in each subject by the topper of the class.
- v) Draw the doughnut chart of the data as in (iv)
- vi) Enter the S.No. of a student and find out the Grade of the student using VLOOKUP.
- vii) Extract all records where name
  - a) Begins with "A"
  - b) Contains "A"
  - c) Ends with "A"

### **Presentation Tool**

1. Make a presentation of College Education System using
  - a. Blank Presentation
  - b. From Design Template
  - c. From Auto Content Wizard
2. Make a presentation on "Wild Life" and apply the following:
  - a. Add audio and video effects
  - b. Apply various Color Schemes
  - c. Apply various animation schemes.
  - d. Apply Slide Show

### **Reference Books**

1. Introduction to Information Technology, ITL solutions Ltd, Pearson Education
2. A Text Book of Information Technology by V. Rajaraman.
3. Introduction to Information Technology, ITL solutions Ltd, Pearson Education  
Fundamentals of Computer by V. Rajaraman.
4. P. K. Sinha & Priti Sinha, "Computer Fundamentals", BPB Publications, 2007.
5. Dr. Anita Goel, Computer Fundamentals, Pearson Education, 2010.
6. T. Budd, Exploring Python, TMH, 1st Ed, 2011
7. Allen Downey, Jeffrey Elkner, Chris Meyers, How to think like a computer scientist: learning with Python, Freely available online. 2012
8. Sushila Madan, Introduction to Essential tools, JBA, 2009.
9. Anita Goel, Computer Fundamentals, Pearson, 2012

# **CSC-HG-2016: Programming in C++**

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Introduction to C++ (3 Lectures)**

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

## **UNIT 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, Formatted and Console I/O (cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.hetc).

## **UNIT 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)

## **UNIT 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

## **UNIT 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.

## **UNIT 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

#### **UNIT 7: Memory Allocation in C++**

**(3 Lectures)**

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

#### **UNIT 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), Macros

#### **UNIT 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.

#### **UNIT 8: 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)

#### **UNIT 9: 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.

### **Practical / Lab work to be performed**

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

### **Reference Books**

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

2. Bjarne Stroustrup, "The C++ Programming Language", 4<sup>th</sup> Edition, Addison-Wesley , 2013.
3. Bjarne Stroustrup, "Programming -- Principles and Practice using C++", 2<sup>nd</sup> 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.
6. John R. Hubbard, "Programming with C++", Schaum's Series, 2nd Edition, 2000.
7. Andrew Koenig, Barbara E. Moo, "Accelerated C++", Published by Addison-Wesley , 2000.
8. Scott Meyers, "Effective C++", 3rd Edition, Published by Addison-Wesley, 2005.
9. Harry, H. Chaudhary, "Head First C++ Programming: The Definitive Beginner's Guide", First Create space Inc, O-D Publishing, LLC USA.2014
10. Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.
11. Stanley B. Lippman, Josee Lajoie, Barbara E. Moo, "C++ Primer", Published by Addison-Wesley, 5th Edition, 2012



# CSC-HG-2026: Database Management System (CSC-RC-2016)

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Introduction to Database Management Systems (10 Lectures)**

Characteristics of database approach, data models, DBMS architecture and data independence

## **UNIT 2: Entity Relationship and Enhanced ER Modeling (15 Lectures)**

Entity types, relationships, SQL-99: Schema Definition, constraints and object modeling

## **UNIT 3: Relational Data Model (15 Lectures)**

Basic concepts, relational constraints, relational algebra, SQL queries

## **UNIT 4: Database Design (20 Lectures)**

ER and EER to relational mapping, functional dependencies, normal forms up to third normal form

### **Practical / Lab work to be performed**

**Note: MyAccess/MySQL may be used.**

The following concepts must be introduced to the students:

#### **DDL Commands**

┆ Create table, alter table, drop table

#### **DML Commands**

┆ Select, update, delete, insert statements

┆ Condition specification using Boolean and comparison operators (and, or, not, =, <, >, <=, >=, <=)

┆ Arithmetic operators and aggregate functions (Count, sum, avg, Min, Max)

┆ Multiple table queries (join on different and same tables)

┆ Nested select statements

┆ Set manipulation using (any, in, contains, all, not in, not contains, exists, not exists, union, intersect, minus, etc.)

┆ Categorization using group by.....having

┆ Arranging using order by



- department and within each department ordered alphabetically by employee first name.
18. Select the names of employees whose salary does not match with salary of any employee in department 10.
  19. Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee.
  20. Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston or Stafford.
  21. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary. Display with proper headings.
  22. Find the sum of the salaries and number of employees of all employees of the 'Marketing' department as well as the maximum salary, the minimum salary, and the average salary in this department.
  23. Select the names of employees whose salary is greater than the average salary of all employees in department 10.
  24. For each department, retrieve the department number, the number of employees in the department, and their average salary.
  25. For each project, retrieve the project number, the project name, and the number of employees who work on that project.
  26. Change the location and controlling department number for all projects having more than 5 employees to 'Bellaire' and 6 respectively.
  27. For each department having more than 10 employees, retrieve the department no, no of employees drawing more than 40,000 as salary.
  28. Insert a record in Project table which violates referential integrity constraint with respect to Department number. Now remove the violation by making necessary insertion in the Department table.
  29. Delete all dependents of employee whose ssn is '123456789'.
  30. Delete an employee from Employee table with ssn = '12345'( make sure that this employee has some dependents, is working on some project, is a manager of some department and is supervising some employees). Check and display the cascading effect on Dependent and Works on table. In Department table MGRSSN should be set to default value and in Employee table SUPERSSN should be set to NULL
  31. Perform a query using alter command to drop/add field and a constraint in Employee table.

## Reference Books

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems, 6<sup>th</sup> Edition, Pearson Education, 2010.
2. R. Ramakrishanan, J. Gehrke, Database Management Systems, 3<sup>rd</sup> Edition, McGraw-Hill, 2002.
3. A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts, 6th Edition, McGraw Hill, 2010.
4. R. Elmasri, S.B. Navathe, Database Systems Models, Languages, Design and application Programming, 6 Edition, Pearson Education, 2013.

# **CSC-HG-2036: Introduction to Programming in C (TCA-RC-2016)/ (TIT-RC-2016)**

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Introduction to C (20 Lectures)**

Elementary data types, variables, constants and identifiers, Integer, character floating point and string constants, variable declarations. Syntax and semantics, reserved word, Initialization of variable during declarations Constant data types, Expression in C, precedence and associativity of C operators, unary, binary and ternary operators, C arithmetic operators, assignment operators, relational operators, logical operators and bit-wise operators, L-value and R-value, Side effects of operators, Expression statement, Conditional Statement-if, if-else, switch, Iterative Statement-while, do-while, for Other Statement –break, continue, goto, return, null Statement, block Statement. Function: function declaration. Calling a function, Parameters –Call by value, Call by reference and its absence in C, Recursion and how it works, Cast and sizeof operator, Automatic type Conversion

## **UNIT 2: Arrays and pointers (20 Lectures)**

Storage classes: Automatic, External, Static, Register, Scope and lifetime of variables, Arrays and pointers and corresponding operators, Pointer arithmetic, Programs using arrays and pointers like sum, average, minimum, maximum of an array of numbers, Add and delete an element of an array, Merge two sorted arrays, String manipulation programs like addition, subtraction, multiplication and their combinations, Sum of rows, columns, and diagonal elements of a matrix. Transpose of a matrix, linear search, binary search, Selection sort and bubble sort.

## **UNIT 3: Structures and Files (20 Lectures)**

Structure – declaration and use, Structure member resolution and structure pointer member resolution operators, Programs to show the use of structure, Standard C library. Files in C—opening, closing, reading and writing of files, seeking forward and backward, simple examples of file handling programs.

## **Practical / Lab work to be performed**

(At least 20 programming assignments have to be done by each student from the following list. The assignments should be selected in such a way that all the features of C language are covered.)

1. Write a program to convert a given temperature value from Fahrenheit scale to Centigrade scale and vice versa.
2. Write a program to display ASCII value of a character.
3. Write a program to check whether a number is perfect or not.
4. Write a program to find out the biggest of three numbers using nested if.
5. A company insures its drivers if either of the following conditions are satisfied
  - Driver is married.

- Driver is an unmarried, male and above 30 years of age.
- Driver is unmarried, female and above 25 years of age.

Write a program to decide if a driver is to be insured using logical operators.

- Write a program to read a list of positive integers terminated by -1 and display the odd and even numbers separately and also their respective counts.
- Write a program to read values of n and x and print the value of y using switch case where
  - $y=n+x$  when  $n=1$
  - $y=1+x/n$  when  $n=2$
  - $y= n+3x$  when  $n=3$
  - $y=1+nx$  when  $n>3$  or  $n<1$ .
- Write a program to n values of sales and then calculate the commission on sales amount where the commission is calculated as follows:
  - If sales  $\leq$  Rs.500, commission is 5%.
  - If sales  $> 500$  but  $\leq 2000$ , commission is Rs 35 plus 10% above Rs 500.
  - If sales  $> 2000$  but  $\leq 5000$ , commission is Rs 185 plus 12% above Rs.2000.
  - If sales  $> 5000$ , commission is 12.5%.
- Write a program to find out minimum, maximum, sum and average of n numbers without using array.
- Write a program to find out the roots of a quadratic equation. Use proper testing to find checks for real and complex roots.
- Write a program to print the digits of a number in words. ( e.g. if a number 841 is entered through the keyboard your program should print “Eight Four One”.)
- Write a program to print the PASCAL Triangle up to the  $n^{\text{th}}$  row where n is an input to the program.
- Write a function to return the HCF of two positive integers. Write a main function to read two positive integers and print their HCF and LCM by using the above function.
- Write a program to convert a decimal number into binary number using function.
- Write a program to display the result of sine series using function.
- Write a program to find the sum of the following series  
 $1+x-x^3/3!+x^5/5!-x^7/7!+\dots$  corrected up to the 3 decimal place.
- Write a program to read n numbers in a sorted array and insert a given element in a particular position
- Write functions to compute the factorial of a number using both recursive and non-recursive procedure.
- Write a program to print the values of  ${}^n C_r$  and  ${}^n P_r$  for given positive integers  $n \geq r > 0$ . Use a function fact(n) to return the factorial of a non-negative integer n.  
 ${}^n C_r = n!/r!(n-r)!$                        ${}^n P_r = n!/(n-r)!$
- Write a program to display the first n Fibonacci numbers using function.
- Write a program to display the prime numbers within a given range. Write a function to check whether a given integer is prime or not and use it.
- Write a program to Multiply two matrices using function
- Write a program to display the upper Triangle and lower Triangle of a given square matrix using function.
- Write a function to check if a given square matrix is symmetric or not. Write a main function to implement it.

25. Write a program to read a  $m \times n$  matrix and calculate the Row sum and Column Sum of the matrix
26. Write a function to read in an integer and print the representation of the number using the sign and magnitude representation scheme using 8 bits. The program should check for overflow/under flow conditions. The left most bit is to be used as the sign bit.
27. Write a program to merge two sorted arrays.
28. Write a program to count the number of vowels in a string.
29. Write a program to concatenate two strings using function (without using library function).
30. Write a program to convert a string from upper case to lower case and vice versa.
31. Write a program to swap two numbers using function (pass the pointers).
32. Write a program to sort  $n$  number of strings in ascending order using pointer.
33. Write a program using pointers to copy a string to another string variable (without using library function).
34. Declare a structure of a student with details like roll number, student name and total marks. Using this, declare an array with 50 elements. Write a program to read details of  $n$  students and print the list of students who have scored 75 marks and above.
35. Create a structure to store the following information of employees.
  - a. Employee's number, name, pay and date of joining.
 It has been decided to increase the pay as per the following rules:
 

Pay $\leq$ Rs.3000	: 20% increase
Pay $\leq$ Rs.6000 but $>$ Rs.3000	: 15% increase
Pay $>$ Rs.6000	: no increase

 Write a program to implement the above structure.
36. Write a program to read a text file and count the number of vowels in the text file.
37. Write a program to copy a text file to another file.

## Reference Books

1. Programming with C, B.S. Gottfried, Tata Mc-Graw Hill.
2. The C Programming Language, B.W. Kernighan and D.M.Ritchie, PHI
3. Herbert Schild, "The complete reference to C++", Osborn McGraw Hill
4. R. Lafore, "Object Oriented Programming using C++", Galgotia Publications
5. Ian Graham, "Object Oriented Methods", Addison Wesley.
6. Programming in ANSI C, E.Balagurusamy, Tata McGraw – Hill

# CSC-HG-3016: Programming in Python

04 Lectures, 04 Practical, Credits 6 (4+2)

Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Planning the Computer Program (4 Lectures)**

Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation

## **UNIT 2: Techniques of Problem Solving (6 Lectures)**

Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming

## **UNIT 3: Overview of Programming (4 Lectures)**

Structure of a Python Program, Elements of Python

## **UNIT 4: Introduction to Python (6 Lectures)**

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

## **UNIT 5: Creating Python Programs (8 Lectures)**

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

## **UNIT 6: Iteration and Recursion (8 Lectures)**

Conditional execution, Alternative execution, Nested conditionals, the return statement, Recursion, Stack diagrams for recursive functions, Multiple assignment, the while statement, Tables, Two-dimensional tables

## **UNIT 7: Strings and Lists (8 Lectures)**

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

## **UNIT 8: Object Oriented Programming (4 Lectures)**

Introduction to Classes, Objects and Methods, Standard Libraries

## **UNIT 9: Data Structures (6 Lectures)**

Arrays, list, set, stacks and queues.

## **UNIT 10: Searching and Sorting (6 Lectures)**

Linear and Binary Search, Bubble sort, Selection sort and Insertion sort.

## Practical / Lab work to be performed

(N.B: Student has to perform **any eight** of the following experiments)

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 sines, 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  $1 + 1/1! + 1/2! + 1/3! + \dots + 1/n$
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 is 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 a number is Armstrong or not
11. Write a program to check whether the input number is even or odd.
12. Write a program to print all even number between a range(for example between 1 and 100).
13. Write a program to print all prime number between a range(for example between 1 and 100).
14. Write a program to compare three numbers and print the largest one.
15. Write a program to print factors of a given number.
16. Write a method to calculate GCD of two numbers.
17. Write a program to create Stack Class and implement all its methods. (Use Lists).
18. Write a program to create Queue Class and implement all its methods. (Use Lists)
19. Write a program to implement linear and binary search on lists.
20. Write a program to sort a list using insertion sort and bubble sort and selection sort.

## Reference Books

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. Python Tutorial/Documentation [www.python.org](http://www.python.org) 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/>



# **CSC-HG-3026: Computer Networks and Internet Technologies**

04 Lectures, 04 Practical, Credits 6 (4+2)

Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Computer Networks (6 Lectures)**

Introduction to computer network, data communication, components of data communication, data transmission mode, data communication measurement, LAN, MAN, WAN, wireless LAN, internet, intranet, extranet

## **UNIT 2: Network Models (8 Lectures)**

Client/ server network and Peer-to-peer network, OSI, TCP/IP, layers and functionalities

## **UNIT 3: Transmission Media (4 Lectures)**

Introduction, Guided Media: Twisted pair, Coaxial cable, Optical fiber. Unguided media: Microwave, Radio frequency propagation, Satellite.

## **UNIT 4: LAN Topologies (2 Lectures)**

Ring, Bus, Star, Mesh and Tree topologies

## **UNIT 5: Network Devices (2 Lectures)**

NIC, Repeaters, Hub, Bridge, Switch, Gateway and Router

## **UNIT 6: Internet Terms (2 Lectures)**

Web page, Home page, Website, Internet Browsers, URL, Hypertext, ISP, Web server, download and upload, online and offline

## **UNIT 7: Internet Applications (6 Lectures)**

WWW, Telnet, FTP, E-mail, Social Networks, Search Engines, Video Conferencing, E-Commerce, M-Commerce, VOIP, Blogs

## **UNIT 8: Introduction to Web Design (14 Lectures)**

Introduction to HyperText Markup Language (html), Document type definition, Creating Web pages, Lists, Hyperlinks, Tables, Web forms, Inserting images, Frames, Hosting options and Domain name registration, Customized Features: Cascading Style Sheet (CSS) for text formatting and other manipulations

## **UNIT 9: JavaScript Fundamentals (14 Lectures)**

Data types and variables, functions, methods and events, controlling program flow, JavaScript object model, built-in objects and operators.

## **Practical / Lab work to be performed**

**Practical exercises based on concepts listed in theory using HTML.**

(N.B: Student has to perform **any five** of the following experiments)

1. Create HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.
2. Create HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking
3. Create HTML document that contains a table.
4. Create Form with Input Type, Select and Text Area in HTML.
5. Create an HTML containing Roll No., student's name and Grades in a tabular form.
6. Create an HTML document (having two frames) which will appear as follows:

About	This frame would show the contents according to the link clicked by the user on the left frame.
Department 1	
Department 2	
Department 3	

7. Create an HTML document containing horizontal frames as

follows: Department Names (could be along with Logos)
Contents according to the Link clicked

8. Create a website of 6 – 7 pages with different effects as mentioned in above problems.
9. Create HTML documents (having multiple frames) in the following three formats:

Frame1
Frame 2

Frame1	
Frame 2	Frame 3

10. 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

---

### List of Practical's using JavaScript

(N.B: Student has to perform **any five** of the following experiments)

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 numbers.
5. A person deposits Rs 1000 in a fixed account yielding 5% interest. Compute the amount in the account at the end of each year for n years.
6. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.

### Reference Books

1. Andrew S. Tanenbaum, David J. Wetherall Computer Networks (5th Edition), PHI, 2010
2. B. A. Forouzan, Data Communication and Networking , TMH,2003.
3. D.R. Brooks, An Introduction to HTML and Javascript for Scientists and Engineers, Springer W. Willard,2009
4. HTML A Beginner's Guide, Tata McGraw-Hill Education, 2009.
5. J. A. Ramalho, Learn Advanced HTML 4.0 with DHTML, BPB Publications, 2007

## **CSC-HG-3036: Operating Systems (CSC-RC-3016)/( TCA-RC-3016)**

4 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

### **UNIT 1: Introduction (2 Lectures)**

System Software, Resource Abstraction, OS strategies

### **UNIT 2: Types of Operating Systems (2 Lectures)**

Multiprogramming, Batch, Time Sharing, Single user and Multiuser, Process Control & Real Time Systems

### **UNIT 3: Operating System Organization (10 Lectures)**

Factors in operating system design, basic OS functions, implementation consideration, process modes, methods of requesting system services – system calls and system programs

### **UNIT 4: Process Management (15 Lectures)**

System view of the process and resources, initiating the OS, process address space, process abstraction, resource abstraction, process hierarchy, Thread model

### **UNIT 5: Scheduling (12 Lectures)**

Scheduling Mechanisms, Strategy selection, non-pre-emptive and pre-emptive strategies

### **UNIT 6: Memory Management (12 Lectures)**

Mapping address space to memory space, memory allocation strategies, fixed partition, variable partition, paging, virtual memory

### **UNIT 7: Introduction to Shell Programming (7 Lectures)**

Shell introduction and Shell Scripting, What is shell and various type of shell, Various editors present in linux, Different modes of operation in vi editor, What is shell script, Writing and executing the shell script , Shell variable (user defined and system variables), 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)

### **Practical / Lab work to be performed**

**Note: Following exercises can be performed using Linux or Unix**

1. Usage of following commands:  
ls, pwd, tty, cat, who, who am I, rm, mkdir, rmdir, touch, cd.
2. Usage of following commands:  
cal, cat(append), cat(concatenate), mv, cp, man, date.
3. Usage of following commands:  
chmod, grep, tput (clear, highlight), bc.
4. Write a shell script to check if the number entered at the command line is prime or not.

5. Write a shell script to modify “cal” command to display calendars of the specified months.
6. Write a shell script to modify “cal” command to display calendars of the specified range of months.
7. Write a shell script to accept a login name. If not a valid login name display message – “Entered login name is invalid”.
8. Write a shell script to display date in the mm/dd/yy format.
9. Write a shell script to display on the screen sorted output of “who” command along with the total number of users.
10. Write a shell script to display the multiplication table any number,
11. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
12. Write a shell script to find the sum of digits of a given number.
13. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
14. Write a shell script to find the LCD (least common divisor) of two numbers.
15. Write a shell script to perform the tasks of basic calculator.
16. Write a shell script to find the power of a given number.
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.
20. Program to show the pyramid of special character “\*”.

## **Reference 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 , 5 Edition, Prentice Hall of India. 2008.
5. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992

## **CSC-HG-3046: ICT Hardware (TIT-RC-3016)**

04 Lectures, 04 Practical, Credits 6 (4+2)

Theory: 60 Lectures      Practical: 60 Lectures

### **UNIT I (12 Lectures)**

Evolution of computer system, Modern computer, Classification of computer, Personal Computer hardware: Monitor, Keyboard, Mouse, Scanner, printer, speaker

### **UNIT II (12 Lectures)**

Hard Disk Drive: logical structure and file system, FAT, NTFS. Hard disk tools: Disk cleanup, error checking, de fragmentation, scanning for virus, formatting, installing additional HDD, New trends in HDD, Floppy Disk Drive

### **UNIT III (12 Lectures)**

Optical Media, CDROM, drive speed, buffer, cache, CD-r, CD-RW, DVD ROM, DVD technology, preventive maintenance for DVD and CD drives, New Technologies, Driver installation, Writing and cleaning CD and DVD

### **UNIT IV (12 Lectures)**

Processor: Intel processor family. Latest trends in processor, Motherboard, Sockets and slots, power connectors, Peripheral connectors, Bus slots, USB, pin connectors, Different kinds of motherboards, RAM, different kinds of RAM. RAM up gradation, Cache and Virtual Memory (only concept will be given without going into implementation details)

### **UNIT V (12 Lectures)**

SMPS, BIOS, Network Interface Card, network cabling, I/O Box, Switches, RJ 45 connectors, Patch panel, Patch cord, racks, IP address.

## **Practical / Lab work to be performed**

### **Objectives**

The Practical introduces the students to a personal computer and its basic peripherals, the process of assembling a personal computer, installation of system software like Windows OS, LINUX OS, device drivers. Basic system administration in Linux which includes: Basic Linux commands in bash, Create hard and symbolic links, Text processing, Using wildcards In addition hardware and software level troubleshooting process, tips and tricks would be covered.

Different ways of hooking the PC on to the network and internet from home and workplace and effectively usage of the internet, Configuring the TCP/IP setting, Usage of web browsers, email, newsgroups and discussion forums would be covered. In addition, awareness of cyber hygiene, i.e., protecting the personal computer from getting infected with the viruses, worms and other cyber attacks would be introduced. The students should perform the following tasks.

- **Task 1:** Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.
- **Task 2:** Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva.
- **Task 3:** Every student should individually install MS windows on the personal computer. They should learn how to format and partition a hard disk. Lab instructor should verify the installation and follow it up with a Viva.
- **Task 4:** Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both windows and Linux. Lab instructors should verify the installation and follow it up with a Viva
- **Task 5:** Basic commands in Linux
- **Task 6:** Hardware Troubleshooting: Students have to be given a PC which does not boot due to improper assembly or defective peripherals. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva
- **Task 7:** Software Troubleshooting: Students have to be given a malfunctioning CPU due to system software problems. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.
- **Task 8:** The test consists of various systems with Hardware / Software related troubles, formatted disks without operating systems. Installation of antivirus software, configure their personal firewall and windows update on their computer. Then they need to customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

## Reference Books

- 1 Comdex; Hardware and Networking Course Kit; Dream Tech press
- 2 PC hardware: A beginners Guide; Ron Gilster; Tata Mc Graw Hill.
- 3 Computer Fundamentals, Anita Goel, Pearson, 2010.
- 4 Vikas Gupta; Comdex" Hardware and Networking Course Kit; DreamTech press.
- 5 Ron Gilster; PC hardware: A beginners Guide; Tata McGraw Hill. (First edition)

# CSC-HG-4016: Programming in Visual Basic / Gambas

04 Lectures, 04 Practical, Credits 6 (4+2)

Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: GUI Environment**

**(4 Lectures)**

Introduction to Graphical User Interface (GUI), Programming Language (Procedural, Object oriented, Event driven), the GUI environment, compiling, debugging, and running the programs

## **UNIT 2: Controls**

**(8 Lectures)**

Introduction to controls textboxes, frames, check boxes, option buttons, images, setting borders and styles, the shape control, the line control, slider, Tree view control, working with multiple controls and their properties, designing the user interface, keyboard access, tab controls, default & cancel property, coding for controls, Data grid, Flexi grid.

## **UNIT 3: Operations**

**(4 Lectures)**

Data types, constants, named & intrinsic, declaring variables, scope of variables, val function, arithmetic operations, formatting data.

## **UNIT 4: Decision Making**

**(8 Lectures)**

If statement, comparing strings, compound conditions (and, or, not), nested if statements, case structure, using if statements with option buttons & checkboxes, displaying message in message box, testing whether input is valid or not.

## **UNIT 5: Modular programming**

**(6 Lectures)**

Menus, sub-procedures and sub-functions defining creating and modifying a menu, using common dialog box, creating a new sub-procedure, passing variables to procedures, passing argument by value or by reference, writing a function/ procedure.

## **UNIT 6: Forms Handling**

**(6 Lectures)**

Multiple forms creating, adding, removing forms in project, hide, show method, load, unload statement, me keyword, referring to objects on a different Forms

## **UNIT 7: Iteration Handling**

**(7 Lectures)**

Do/loops, for/next loops, using msgbox function, using string Function

## **UNIT 8: Arrays and Grouped Data Control**

**(10 Lectures)**

Arrays - 1-dimension arrays, initializing an array using for each, user-defined data types, accessing information with user-defined data types, using list boxes with array, two dimensional arrays, lists, loops and printing list boxes & combo boxes, filling the list using property window / add item method, clear method, list box properties, removing an item from a list, list box/ combo box operations.



## UNIT 9: Database Connectivity

(10 Lectures)

Database connectivity of forms with back end tool like mysql, populating the data in text boxes, list boxes etc. searching of data in database using forms. Updating/ editing of data based on a criterion.

### Practical / Lab work to be performed

(N.B: Student has to perform **any six** of the following experiments)

1. Print a table of numbers from 5 to 15 and their squares and Cubes.
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 numbers.
5. A person deposits Rs. 1000 in a fixed account yielding 5% interest. Complete the amount in the account at the end of each year for n years.
6. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.
7. Read n numbers. Count the number of negative numbers, positive numbers and zeroes in the list.use arrays.
8. Read a single dimension array. Find the sum and average of these numbers.
9. Read a two dimension array. Find the sum of two 2D Array.
10. Create a database Employee and Make a form in VB 6.0 to allow data entry to Employee Form with the following command buttons:

#### Employee Form

Employee Name:	<input type="text"/>
Employee Id:	<input type="text"/>
Date of Joining:	<input type="text"/>
Designation:	<input type="text"/>
Department:	<input type="text"/>
Address:	<input type="text"/>
Basic Pay:	<input type="text"/>

### Reference Books

1. Programming in Visual Basic 6.0 by Julia Case Bradley, Anita C. Millispangh (Tata Mcgraw Hill Edition 2000 (Fourteenth Reprint 2004))

# **CSC-HG-4026: Information Security and Cyber Laws**

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

## **UNIT 1: Introduction (8 Lectures)**

Information Security- History, overview, Definitions, Basic principles, Key Concepts, Security threats and responses, Security classification for information.

## **UNIT 2: Digital Crime (4 Lectures)**

Overview of Digital crime/ Cyber crime, criminology related to computer systems and cyber space

## **UNIT 3: Information Gathering Techniques (8 Lectures)**

Tools of the attacker, information and cyber warfare, scanning and spoofing, password cracking, Sniffing, malicious software, Session Hijacking

## **UNIT 4: Risk Analysis and Threat (10 Lectures)**

Risk analysis, Risk Management, Risk assessment, Key principles of Computer security, Security Policies, Central Mechanisms, Authentication and Authorization techniques, Data Protection, Access Control, Digital Evidence, Computer forensics and Incident Response Plans.

## **UNIT 5: Introduction to Cryptography and Applications (10 Lectures)**

Concepts of cryptography, Types of cryptography, Digital Signature, Digital Signature Certified Codifying techniques, Ciphers (Caesar Cipher, Rail-Fence Cipher), Application of Cryptography

## **UNIT 6: Safety Tools and Issues (10 Lectures)**

Security Governance, Firewalls, Logging and Intrusion Detection systems, Windows and windows XP / NT security, Unix/Linux security, ethics of hacking and cracking

## **UNIT 7: Cyber laws to be covered as per IT 2008 (10 Lectures)**

Chapter 1: Definitions

Chapter 2: Digital Signature And Electronic Signature

[Section 43] Penalty and Compensation for damage to computer, computer system, etc.

[Section 65] Tampering with Computer Source Documents

[Section 66 A] Punishment for sending offensive messages through communication service, etc.

[Section 66 B] Punishments for dishonestly receiving stolen computer resource or communication device

[Section 66C] Punishment for identity theft

[Section 66D] Punishment for cheating by personating by using computer resource

[Section 66E] Punishment for violation of privacy

[Section 66F] Punishment for cyber terrorism

[Section 67] Punishment for publishing or transmitting obscene material in electronic form

[Section 67A] Punishment for publishing or transmitting of material containing sexually explicit act, etc. in electronic form  
[Section 67B] Punishment for publishing or transmitting of material depicting children in sexually explicit act, etc. in electronic form

[Section 72] Breach of confidentiality and privacy

### **Practical / Lab work to be performed**

**(N.B:** Student has to perform **any four** of the following experiments)

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.

### **Reference Books**

1. M. Merkow, J. Breithaupt, Information Security Principles and Practices, Pearson Education.2005
2. G.R.F. Snyder, T. Pardoe, Network Security, Cengage Learning, 2010
3. A. Basta, W.Halton, Computer Security: Concepts, Issues and Implementation, Cengage Learning India, 2008

## **CSC-HG-4036: Web and E-Commerce Technologies**

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

### **UNIT 1: An introduction to Electronic commerce (10 Lectures)**

What is E-Commerce (Introduction And Definition), Main activities E-Commerce, Goals of E-Commerce, Technical Components of E-Commerce, Functions of E-Commerce, Advantages and disadvantages of E-Commerce, Scope of E-Commerce, Electronic Commerce Applications, Electronic Commerce and Electronic, Business models (C2B,C2C, B2B, B2C,B2G,G2B,G2C)

### **UNIT 2: The Internet and WWW (10 Lectures)**

Evolution of Internet, Domain Names and Internet Organization (.edu, .com, .mil, .gov, .net etc.) , Types of Network, Internet Service Provider, World Wide Web, Internet & Extranet, Role of Internet in B2B Application, building own website, Cost, Time, Reach, Registering a Domain Name, Web promotion, Target email, Banner, Exchange, Shopping Bots

### **UNIT 3: Internet Security (10 Lectures)**

Secure Transaction, Computer Monitoring, Privacy on Internet, Corporate Email privacy, Computer Crime( Laws, Types of Crimes), Threats, Attack on Computer System, Software Packages for privacy, Hacking, Computer Virus( How it spreads, Virus problem, virus protection, Encryption and Decryption, Secret key Cryptography, DES, Public Key Encryption, RSA, Authorization and Authentication, Firewall, Digital Signature( How it Works)

### **UNIT 4: Electronic Data Exchange (10 Lectures)**

Introduction, Concepts of EDI and Limitation, Applications of EDI, Disadvantages of EDI, EDI model, Electronic Payment System: Introduction, Types of Electronic Payment System, Payment Types, Value Exchange System, Credit Card System, Electronic Fund Transfer, Paperless bill, Modern Payment Cash, Electronic Cash

### **UNIT 5: Planning for Electronic Commerce (10 Lectures)**

Planning Electronic Commerce initiates, Linking objectives to business strategies, Measuring cost objectives, Comparing benefits to Costs, Strategies for developing electronic commerce web sites

### **UNIT 6: Internet Marketing (10 Lectures)**

The PROS and CONS of online shopping, The cons of online shopping, Justify an Internet business, Internet marketing techniques, The E-cycle of Internet marketing, Personalization e-commerce.

## **Practical / Lab work to be performed**

(N.B: Student has to perform **any four** of the following experiments)

### **Web and E- Commerce Technologies LAB (based on the following topics):**

1. HyperText Markup Language (HTML): structural setup; page layout; text manipulation; special characters; images; links. Intermediate: image maps; tables; frames, forms; meta tags; web forms.
2. Cascading Style Sheets (CSS) : embedding/linking; HTML element selectors; classes; ID selectors, text manipulation; background; borders and spacing; layout; context selectors and grouping, pseudo-classes; pseudo-elements.
3. JavaScript : writing your first script; creating HTML tags; user input and output; loops and tables; payroll calculator, forms and text fields; validating an email address; radio buttons; check boxes; self-grading tests, image rollovers; slide shows; real-time clock; controllable clock; working with cookies.
4. Perl/CGI 10: sample Perl operations; random numbers; lists; dealing four poker hands; time manipulation; subroutines, hash tables; files; string matching, CGI; registration lists; surveys.
5. SQL and regular expressions: Regular expressions: select; where; order by; insert; update; delete, like; aggregate functions; create table; alter table; drop deterministic functions; non-deterministic functions, basics; repeating; positioning.
6. Beginner: between; in; distinct; group by; aliases; table., nested selects; Sound Ex; join;
7. ASP structural setup: response write; retrieving from forms; retrieving from query string; variables; control constructs; subroutines and functions; session state; application variables; server variables; debugging, reading and writing cookies; server-side includes; response object methods; VBScript functions; error handling; debugging, browser details; CDONTS; files; output from a recordset; global.asa; setup instructions for using IIS and ASP. Flash 3 Create Flash movies of moving and interactive objects.

### **Recommended Books**

1. G.S.V.Murthy, E-Commerce Concepts, Models, Strategies- :- Himalaya Publishing House, 2011.
2. Kamlesh K Bajaj and Debjani Nag , E- Commerce , 2005.
3. Gray P. Schneider , Electronic commerce, International Student Edition, 2011,
4. Henry Chan, Raymond Lee, Tharam Dillon, Elizabeth Chang, E Commerce, Fundamentals and Applications, Wiely Student Edition, 2011

## **CSC-HG-4046: Computer System Architecture (CSC-RC-4016)**

4 Lectures, 4 Practical, Credits 6 (4+2)  
Theory: 60 Lectures, Practical: 60 Lectures

### **UNIT 1: Introduction (12 Lectures)**

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

### **UNIT 2: Data Representation and Basic Computer Arithmetic (8 Lectures)**

Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison

### **UNIT 3: Basic Computer Organization and Design (18 Lectures)**

Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt

### **UNIT 4: Central Processing Unit (10 Lectures)**

Register organization, arithmetic and logical micro-operations, stack organization, micro programmed control

### **UNIT 5: Programming the Basic Computer (8 Lectures)**

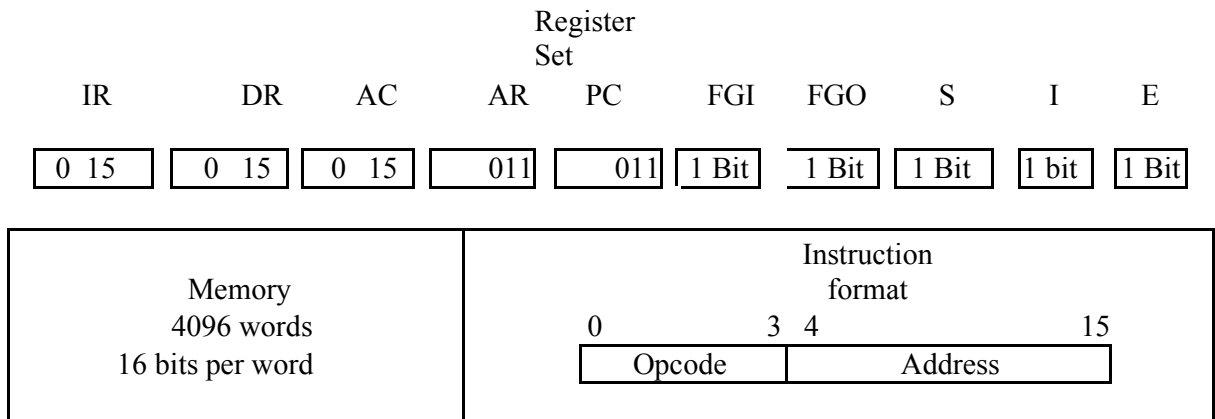
Instruction formats, addressing modes, instruction codes, machine language, assembly language, input output programming

### **UNIT 6: Input-output Organization (4 Lectures)**

Peripheral devices, I/O interface, Modes of data transfer, direct memory access

### **Practical / Lab work to be performed**

1. Create a machine based on the following architecture:



Basic Computer Instructions

Memory Reference		Register Reference		Input-Output		
Symbol	Hex	Symbol	Hex	Symbol	Hex	
AND	0xxx	Addressing	CLA	E800	INP	F800
STA_I	7xxx		SZE	E002		F40
BUN_I	9xxx		HLT	E001		
BSA_I	Bxxx					
ISZ_I	Dxxx					

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

Initialize the contents of AC to (A937)<sub>16</sub>, that of PC to (022)<sub>16</sub> 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**

<b>2</b>	<b>4</b>	<b>15</b>
code	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

**Reference 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. Digital Design, M.M. Mano, Pearson Education Asia, 1979



**CSC-HG-4056: Introduction to DBMS**  
**(TCA-RC-4016)/(TIT-RC-4016)**

04 Lectures, 04 Practical, Credits 6 (4+2)  
Theory: 60 Lectures      Practical: 60 Lectures

**UNIT 1: Introduction to Database Management Systems** **(10 Lectures)**

Characteristics of database approach, data models, DBMS architecture and data Independence

**UNIT 2: Entity Relationship and Enhanced ER Modeling** **(15 Lectures)**

Entity types, relationships, SQL-99: Schema Definition, constraints and object modeling

**UNIT 3: Relational Data Model** **(15 Lectures)**

Basic concepts, relational constraints, relational algebra, SQL queries

**UNIT 4: Database Design** **(20 Lectures)**

ER and EER to relational mapping, functional dependencies, normal forms up to third Normal form

**Practical / Lab work to be performed**

**Note: MyAccess / MySQL may be used.**

The following concepts must be introduced to the students:

**DDL Commands**

‡ Create table, alter table, drop table

**DML Commands**

‡ Select , update, delete, insert statements

‡ Condition specification using Boolean and comparison operators (and, or, not, =, <, >, <=, >=)

‡ Arithmetic operators and aggregate functions(Count, sum, avg, Min, Max)

‡ Multiple table queries (join on different and same tables)

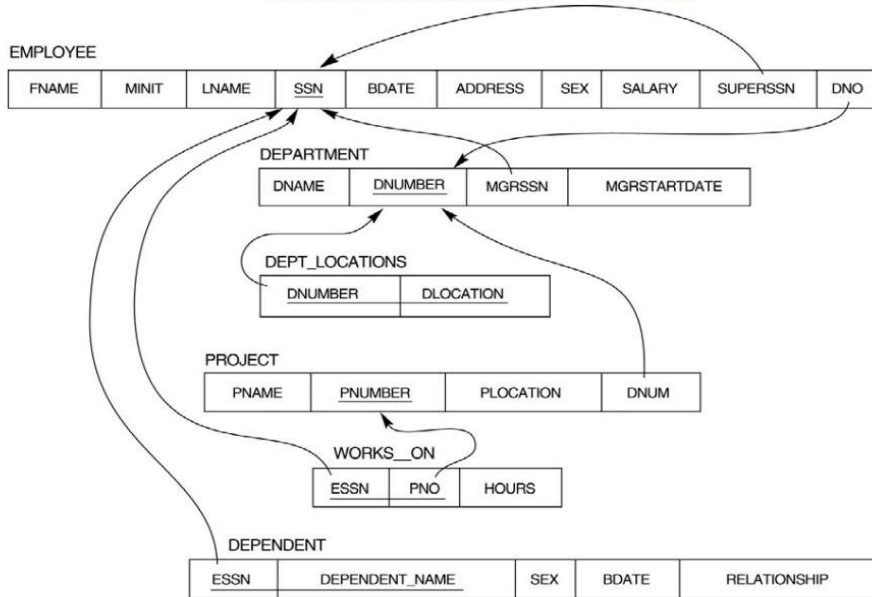
‡ Nested select statements

‡ Set manipulation using (any, in, contains, all, not in, not contains, exists, not exists, union, intersect, minus, etc.)

‡ Categorization using group by.....having

‡ Arranging using order by

### Relational Database Schema - COMPANY



### Questions to be performed on above schema

- A. Create tables with relevant foreign key constraints
- B. Populate the tables with data
- C. Perform the following queries on the database:
  1. Display all the details of all employees working in the company.
  2. Display ssn, lname, fname, address of employees who work in department no 7.
  3. Retrieve the birth date and address of the employee whose name is 'Franklin T. Wong'
  4. Retrieve the name and salary of every employee
  5. Retrieve all distinct salary values
  6. Retrieve all employee names whose address is in 'Bellaire'
  7. Retrieve all employees who were born during the 1950s
  8. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000 (inclusive)
  9. Retrieve the names of all employees who do not have supervisors
  10. Retrieve SSN and department name for all employees
  11. Retrieve the name and address of all employees who work for the 'Research' department
  12. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date
  13. For each employee, retrieve the employee's name, and the name of his or her immediate supervisor
  14. Retrieve all combinations of Employee Name and Department Name
  15. Make a list of all project numbers for projects that involve an employee whose last name is 'Narayan' either as a worker or as a manager of the department that controls the project.
  16. Increase the salary of all employees working on the 'ProductX' project by 15%.  
Retrieve employee name and increased salary of these employees.
  17. Retrieve a list of employees and the project name each works in, ordered by the employee's

- department and within each department ordered alphabetically by employee first name.
18. Select the names of employees whose salary does not match with salary of any employee in department 10.
  19. Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee.
  20. Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston or Stafford.
  21. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary. Display with proper headings.
  22. Find the sum of the salaries and number of employees of all employees of the 'Marketing' department as well as the maximum salary, the minimum salary, and the average salary in this department.
  23. Select the names of employees whose salary is greater than the average salary of all employees in department 10.
  24. For each department, retrieve the department number, the number of employees in the department, and their average salary.
  25. For each project, retrieve the project number, the project name, and the number of employees who work on that project.
  26. Change the location and controlling department number for all projects having more than 5 employees to 'Bellaire' and 6 respectively.
  27. For each department having more than 10 employees, retrieve the department no, no of employees drawing more than 40,000 as salary.
  28. Insert a record in Project table which violates referential integrity constraint with respect to Department number. Now remove the violation by making necessary insertion in the Department table.
  29. Delete all dependents of employee whose ssn is '123456789'.
  30. Delete an employee from Employee table with ssn = '12345' (make sure that this employee has some dependents, is working on some project, is a manager of some department and is supervising some employees). Check and display the cascading effect on Dependent and Works on table. In Department table MGRSSN should be set to default value and in Employee table SUPERSSN should be set to NULL
  31. Perform a query using alter command to drop/add field and a constraint in Employee table.

## Reference Books

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems, 6<sup>th</sup> Edition, Pearson Education, 2010.
2. R. Ramakrishanan, J. Gehrke, Database Management Systems, 3<sup>rd</sup> Edition, McGraw-Hill, 2002.
3. A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts, 6th Edition, McGraw Hill, 2010.
4. R. Elmasri, S.B. Navathe, Database Systems Models, Languages, Design and application Programming, 6 Edition, Pearson Education, 2013.