

# The University of Burdwan



Curriculum for B.Sc. Honours  
in  
Computer Science  
under Semester with CBCS  
w.e.f. 2017 – 2018

## SEMESTER - I

Course code	Course Title	Credit	No of Hours		
			L	T	P
CC - 1	Programming fundamentals using C/C++	4-0-2=6	4	0	4
CC - 2	Computer System Architecture	4-0-2=6	4	0	4
GE-1	from other discipline	6			
AECC-1	Environmental Studies	4	4		
		22			

## SEMESTER - II

Course code	Course title	Credit	No of Hours		
			L	T	P
CC - 3	Programming in Java	4-0-2=6	4	0	4
CC - 4	Discrete Structure	5-1-0=6	5	1	0
GE-2	from other discipline	6			
AECC 2	Communicative English / MIL	2			
		20			

## SEMESTER - III

Course code	Course title	Credit	No of Hours		
			L	T	P
CC - 5	Data Structure	4-0-2=6	4	0	4
CC - 6	Operating Systems	4-0-2=6	4	0	4
CC - 7	Computer Networks	4-0-2=6	4	0	4
GE-3	from other discipline	6			
SEC-1	Programming in PYTHON	2-0-0=2	2	0	0
		26			

#### SEMESTER - IV

Course code	Course title	Credit	No of Hours		
			L	T	P
CC - 8	Design and Analysis of Algorithms	4-0-2=6	4	0	4
CC - 9	Software Engineering	4-0-2=6	4	0	4
CC - 10	Database Management System	4-0-2=6	4	0	4
GE-4	from other discipline	6			
SEC-2	(PHP programming)	2-0-0=2	2	0	0
		26			

#### SEMESTER - V

Course code	Course title	Credit	No of Hours		
			L	T	P
CC - 11	Internet Technologies	4-0-2=6	4	0	4
CC - 12	Theory of Computation	4-0-2=6	4	0	4
	DSE-1	4-0-2=6	4	0	4
	DSE-2	4-0-2=6	4	0	4
		24			

#### SEMESTER - VI

Course code	Course title	Credit	No of Hours		
			L	T	P
CC - 13	Artificial Intelligence	4-0-2=6	4	0	4
CC - 14	Computer Graphics	4-0-2=6	4	0	4
	DSE-3	4-0-2=6	4	0	4
	DSE-4	4-0-2=6	4	0	4
		24			

#### **Core course (14 papers ,credit 6 each (4+2))**

- 1 Programming Fundamentals using C/C++
- 2 Computer System Architecture
- 3 Programming in Java
- 4 Discrete Structure
- 5 Data Structure

6 Operating Systems

7 Computer Networks

8 Design and Analysis of Algorithms

9 Software Engineering

10 Database Management System

11 Internet Technologies

12 Theory of Computation

13 Artificial Intelligence

14 Computer Graphics

### **Discipline specific Elective Papers**

#### **DSE 1**

Microprocessor

or

Digital Image Processing

#### **DSE 2**

Numerical Methods

or

System Programming

#### **DSE 3**

Combinatorial optimization

or

Soft Computing

#### **DSE4**

Data Mining

or

Project Work

## **Skill Enhancement Course**

### **SEC 1**

Programming in Python

Or

R Programming

### **SEC 2**

PHP Programming

Or

UNIX / LINUX Programming

## **GE - 1 to GE - 4 to be offered by Other Disciplines**

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

## **General Electives for other disciplines (GE 1 to GE 4) to be offered by Computer Science Department**

GE 1: Problem solving Using Computer

GE 2: Database management System

GE 3: Operating Systems

GE 4: Computer System Architecture

## Semester - I

### **CC - 1: Programming Fundamentals using C/C++**

**Theory: 60 Lectures**

**Credit: 4**

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

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

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

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

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

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

#### **4. Functions and Arrays (10 Lectures)**

Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments. Creating and Using One Dimensional Arrays ( Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Twodimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays

#### **5. Derived Data Types (Structures and Unions) (3 Lectures)**

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

#### **6. Pointers and References in C++ (7 Lectures)**

Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems with 11 Pointers, Passing pointers as function arguments, Returning a pointer from a function, using arrays as pointers, Passing arrays to functions. Pointers vs. References, Declaring and initializing references, Using references as function arguments and function return values

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

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

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

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

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

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

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

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

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

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

### CC - 1: Programming Fundamentals using C/C++

#### Practical:

Credit: 2

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

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

10. WAP to perform following actions on an array entered by the user: i) Print the even-valued elements  
ii) Print the odd-valued elements  
iii) Calculate and print the sum and average of the elements of array  
iv) Print the maximum and minimum element of array

- v) Remove the duplicates from the array
- vi) Print the array in reverse order

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

11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.

12. Write a program that swaps two numbers using pointers.

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

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

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

16. Write a menu driven program to perform following operations on strings:

- a) Show address of each character in string
- b) Concatenate two strings without using strcat function.
- c) Concatenate two strings using strcat function.
- d) Compare two strings
- e) Calculate length of the string (use pointers)
- f) Convert all lowercase characters to uppercase
- g) Convert all uppercase characters to lowercase
- h) Calculate number of vowels
- i) Reverse the string

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

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

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

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

21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation):

- a) Sum b) Difference c) Product d) Transpose

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

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

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

Write a program which takes input from the user for length, breath and height to test the above class.

25. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
26. Write a program to retrieve the student information from file created in previous question and print it in following format:  
Roll No. Name Marks
27. Copy the contents of one text file to another file, after removing all whitespaces.
28. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
29. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

## Reference Books

1. HerbtzSchildt, "C++: The Complete Reference", Fourth Edition, McGraw Hill.2003
2. BjarneStroustrup, "The C++ Programming Language", 4th Edition, Addison-Wesley , 2013.
3. BjarneStroustrup, "Programming -- Principles and Practice using C++", 2nd Edition, Addison-Wesley 2014.
4. E Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw-Hill Education, 2008. 5. Paul Deitel, Harvey Deitel, "C++ How to Program", 8th Edition, Prentice Hall, 2011.
5. John R. Hubbard, "Programming with C++", Schaum's Series, 2nd Edition, 2000.
6. Andrew Koeni, Barbara, E. Moo, "Accelerated C++", Published by Addison-Wesley , 2000.
7. Scott Meyers, "Effective C++", 3rd Edition, Published by Addison-Wesley, 2005.
8. Harry, H. Chaudhary, "Head First C++ Programming: The Definitive Beginner's Guide", First Create space Inc, O-D Publishing, LLC USA.2014
9. Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.
10. Stanley B. Lippman, JoseeLajoie, Barbara E. Moo, "C++ Primer", Published by AddisonWesley, 5th Edition, 2012



**CC - 2: Computer System Architecture**  
**Theory: 60 Lectures**

**Credit:4**

**1. Introduction (8 lectures)**

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

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

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

**3. Basic Computer Organization and Design (13 lectures)**

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

**4. Central Processing Unit (15 lectures)**

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

**5. Memory Organization (6 lectures)**

Cache memory, Associative memory, mapping.

**6. Input-Output Organization (8 lectures)**

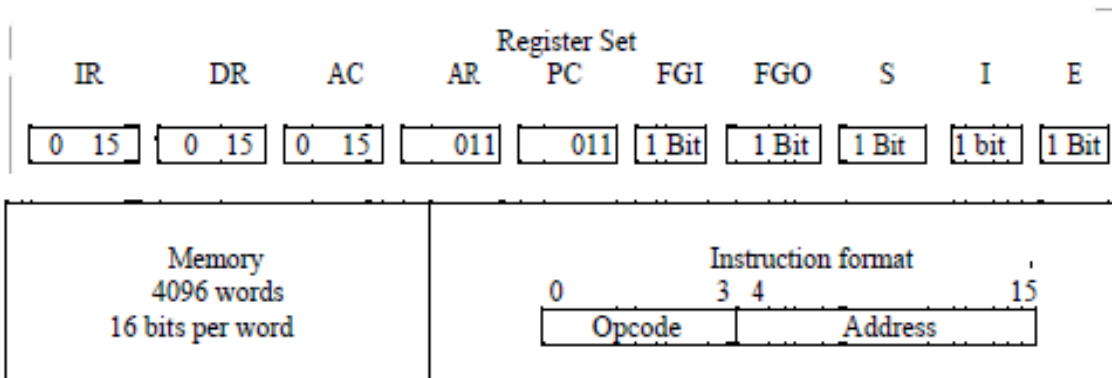
Input / Output: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels.

CC - 2: Computer System Architecture

Practical:

Credit: 2

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	CLA	E800	INP	F800
ADD	2xxx	CLE	E400	OUT	F400
LDA	4xxx	CMA	E200	SKI	F200
STA	6xxx	CME	E100	SKO	F100
BUN	8xxx	CIR	E080	ION	F080
BSA	Axxx	CIL	E040	IOF	F040
ISZ	Cxxx	INC	E020		
AND I	1xxx	SPA	E010		
ADD I	3xxx	SNA	E008		
LDA I	5xxx	SZA	E004		
STA I	7xxx	SZE	E002		
BUN I	9xxx	HLT	E001		
BSA I	Bxxx				
ISZ I	Dxxx				

Optional

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

(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

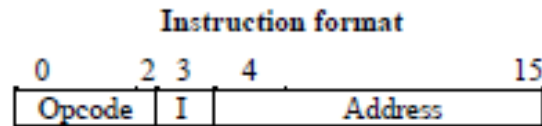
Initialize the contents of AC to  $(A937)_{16}$ , that of PC to  $(022)_{16}$  and E to 1.

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

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

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:



- 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.

## Recommended Books:

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

## Generic Elective (For other disciplines)

### GE-1 : Problem Solving using Computer

**Theory:** 60 Lectures

Credit: 4

#### Computer Fundamentals:

**Introduction to Computers:** Characteristics of Computers, Uses of computers, Types and generations of Computers. (3L)

**Basic Computer Organization:** Units of a computer, CPU, ALU, memory hierarchy, registers, I/O devices. (4L)

#### Planning the Computer Program:

Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation. (3L)

#### Techniques of Problem Solving:

Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming. (4L)

#### Overview of Programming:

Structure of a Python Program, Elements of Python (4L)

#### Introduction to Python:

Python Interpreter, Using Python as calculator, Python shell, Indentation. Atoms, Identifiers and keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit-wise operator, Increment or Decrement operator). (8L)

#### Creating Python Programs:

Input and Output Statements, Control statements (Looping- while loop, for loop , loop Control, Conditional Statement, if...else , Difference between break, continue and pass). (10L)

#### Structures:

Numbers, Strings, Lists, Tuples, Dictionary, Date & Time, Modules, Defining Functions, Exit function, default arguments. (10L)

#### Introduction to Advanced Python:

Objects and Classes, Inheritance, Regular Expressions, Event Driven Programming, GUI Programming. (14L)

## Software Lab using Python:

Credit : 2

### 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/pythonds>

8. <http://www.ibiblio.org/g2swap/byteofpython/read/>

## **Semester - II**

### **CC – 3 : Programming in Java**

**Theory: 60 lectures**

**Credit:4**

#### **1. Introduction to Java**

**(4 Lectures)**

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

#### **2. Arrays, Strings and I/O**

**(8 Lectures)**

Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.

#### **3. Object-Oriented Programming Overview**

**(4 Lectures)**

Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

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

**(14 lectures)**

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

#### **4. Exception Handling, Threading, Networking and Database Connectivity (15 Lectures)**

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

#### **5. Applets and Event Handling (15 Lectures)**

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

### **CC – 3 : Programming in Java**

#### **Practical:**

**Credit: 2**

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of .length in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and StringBuffer class like setCharAt(), setLength(), append(), insert(), 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).
23. Write a program to create URL object, create a URLConnection using the openConnection() method and then use it examine the different components of the URL and content.
24. Write a program to implement a simple datagram client and server in which a message that is typed into the server window is sent to the client side where it is displayed.
25. Write a program that creates a Banner and then creates a thread to scrolls the message in the banner from left to right across the applet's window.
26. Write a program to get the URL/location of code (i.e. java code) and document(i.e. html file).
27. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed(), mouseReleased() and mouseDragged().



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

---

## 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. "Head First Java", Orielly Media Inc. 2nd Edition, 2005.
10. David J. Eck, "Introduction to Programming Using Java", Published by CreateSpace Independent Publishing Platform, 2009.
11. John R. Hubbard, "Programming with JAVA", Schaum's Series, 2nd Edition, 2004.

---

## CC – 4 : Discrete Structures

### Theory: 60 lectures

**Credit:5**

1. **Introduction:** Sets - finite and Infinite sets, uncountably Infinite Sets; functions, relations, Properties of Binary Relations, Closure, Partial Ordering Relations; counting - Pigeonhole Principle, Permutation and Combination; Mathematical Induction, Principle of Inclusion and Exclusion. 15L
2. **Growth of Functions:** Asymptotic Notations, Summation formulas and properties, Bounding Summations, approximation by Integrals. 8L
3. **Recurrences:** Recurrence Relations, generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Recurrence Trees, Master Theorem. 10L
4. **Graph Theory:** 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. 15L

5. **Propositional Logic:** Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory. 12L

### Recommended Books:

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

### CC - 4: Discrete Structures

**Tutorial: 10 lectures**

**Credit: 1**

---

## Generic Elective (For other disciplines)

### GE - 2: Database Management Systems

**Theory: 60 Lectures**

**Credit: 4**

**Introduction to Database Management Systems:** Characteristics of database approach, data models, DBMS architecture and data independence. (10L)

**Entity Relationship and Enhanced ER Modeling:** Entity types, relationships, SQL-99: Schema Definition, constraints, and object modeling. (15L)

**Relational Data Model:** Basic concepts, relational constraints, relational algebra, SQL queries. (15L)

**Database design:** ER and EER to relational mapping, functional dependencies, normal forms up to third normal form. (20 L)

### Software Lab based on Database Management Systems

**Credit: 2**

**Note: MS Access/MySQL may be used.**

The following concepts must be introduced to the students:

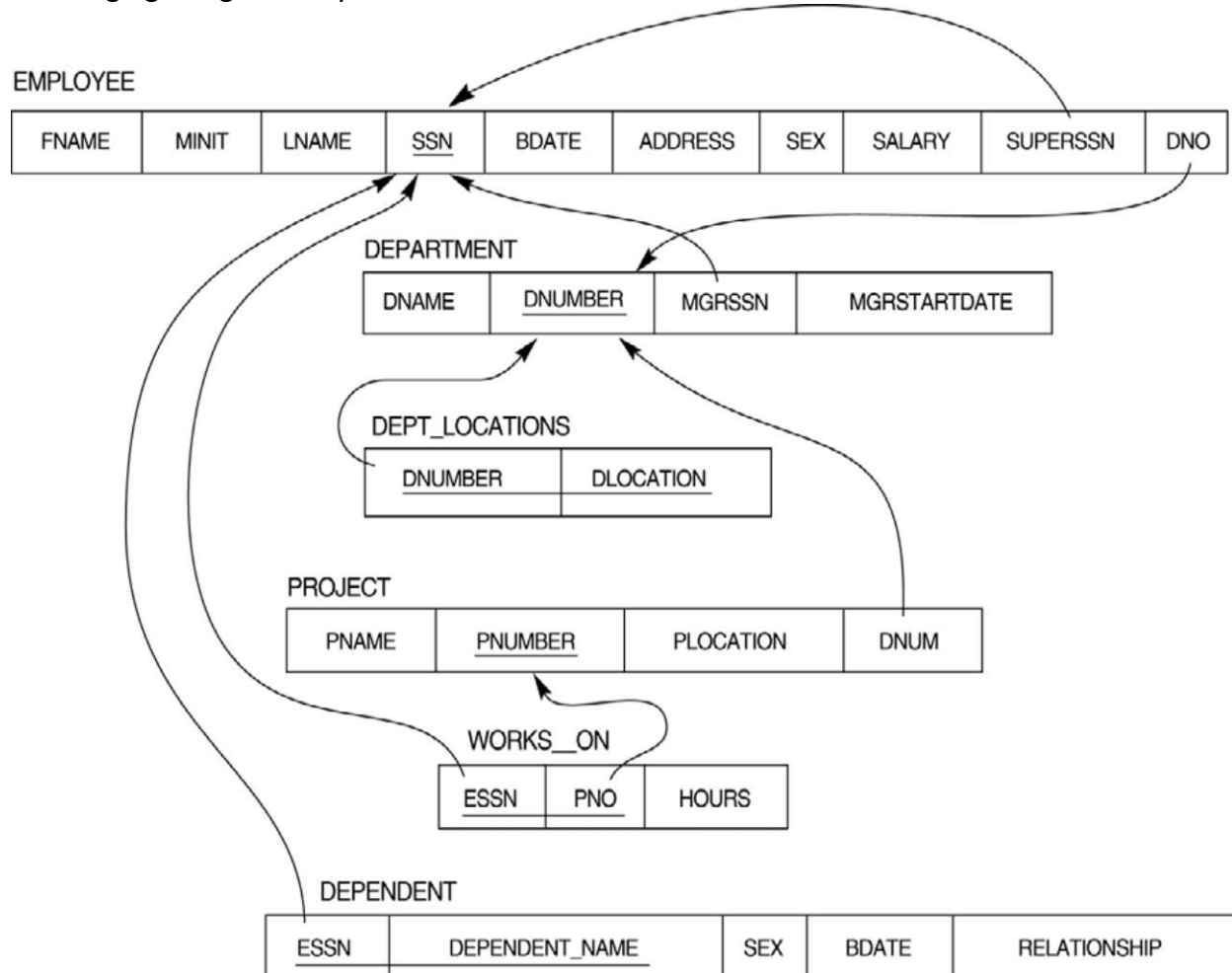
#### DDL Commands

- 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

1. Create tables with relevant foreign key constraints
2. Populate the tables with data
3. 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 birthdate 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 birthdate.
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 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 6th Edition, McGraw Hill, 2010.
4. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education,2013.

=====