Program	Bachelor in Computer Application (BCA)	**Semester - 3**
Type of Course	Core Courses	
Prerequisite	Knowledge of Basic Programming concept of C language.	
Course Objective	This course introduces the fundamental concepts of the Object Oriented Programming using Java. Student will learn to apply basic java constructs, features and library to solve simple problem. Student will execute the implementation of Package, Exception handling mechanism and file handling.	
Effective From A.Y.	2023-24	

Teaching Scheme (Contact Hours)				Examination Scheme				
				Theory Marks		Practical Marks		
Lecture	Tutorial	Lab	Credit	T	T	P	P	Total Marks
3	0	4	5	40	30	20	10	100

***SEE** - Semester End Examination, **T** - Internal Theory, **P** - Internal Practical*

Course Content

T - Teaching Hours | **W** - Weightage

Sr.	Topics	T	W
1	**Introduction to Basic Java Programming** Evolution of Java, Features of JAVA, JDK, JRE, JVM, Byte Code, Installing and configuring JAVA, Basic Structure of Java program, Compiling and Running Java files, Primitive data types, Variables, Type casting, Command line arguments, Operators, Operator precedence	9	20
2	**Control Structure, Function and Array** Control Statements – if, else, nested if, if-else ladder, switch, while, do-while, for, foreach, break and continue statement, Function and its types, Single and Multidimensional Array	8	20
3	**Object Oriented Programming** Object Oriented Programming Concepts like Object, Class and it's relationship, Encapsulation, Inheritance, Polymorphism, Abstraction. Nested class, Anonymous inner class, String, String Buffer, Math class, Wrapper Class	9	20
4	**Inheritance and Abstraction** Method overloading, Constructor, Constructor overloading, this and static keyword, finalize method, Method Overriding, super keyword, final keyword, Abstract class, Interface, Dynamic Method Dispatch	11	20
5	**Package , Exception Handling and IO Programming** Use of package, CLASSPATH, import statement, Access control, Exception and Error, try, catch, throw, throws and finally. Built in exception, Custom exception, File Class, Byte Stream, Character Stream, Exploring classes/interfaces from the java.io package	8	20
	Total	45	100

Suggested Distribution Of Theory Marks Using Bloom's Taxonomy

Level	Remembrance	Understanding	Application	Analyze	Evaluate	Create
Weightage	10	35	55	0	0	0

NOTE : This specification table shall be treated as a general guideline for the students and the teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes

At the end of this course, students will be able to:	
CO1	**use** java constructs, features and libraries for simple problems.
CO2	**apply** the concept of array and control structure to solve the problem.
CO3	**implement** object oriented programming for problem solving.
CO4	**execute** the concept of inheritance and data abstraction.
CO5	**demonstrate** the package, exception handling mechanism and file handling.

Reference Books

1.	**Java Fundamentals** By Herbert Schildt, Dale Skrien | McGraw Hill
2.	**Introduction to Java programming** By Y.Daniel Liang | Pearson
3.	**Programming in Java** By Sachin Malhotra & Saurabh Chaudhary | Oxford University Press India
4.	**Programming with Java** By E Balagurusamy | Tata McGraw Hill

List of Practical

1.	**Demonstration of basics of java programming** 1. Introduction to JDK (Java development kit) and path setting. (A) 2. WAP to print "Welcome to Java". (A) 3. WAP to print your address i) using single print ii) using multiple println. (A) 4. WAP to print addition of 2 number using command line (WITH Integer.parseInt()) (B) 5. WAP to calculate Area of Circle. (C)
2.	**Implementation of Scanner class and other basics of java programming** 1. WAP to print addition of 2 numbers (with Scanner). (A) 2. WAP that reads a number in meters, converts it to feet, and displays the result. (A) 3. WAP to convert temperature from Fahrenheit to Celsius. (B) 4. Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI. Note: - 1 pound=.45359237 Kg and 1 inch=.0254 meters (C)
3.	**Implementation of Control Structure (Part - I)** 1. WAP to find a diameter from given area of circle. (A) 2. WAP to check whether the given number is positive or negative. (A) 3. WAP that prompts the user to enter a letter and check whether a letter is a vowel or consonants. (B) 4. WAP to find out largest number from given three numbers without using Logical Operator.(C) 5. WAP to read marks of five subjects. Calculate percentage and print class accordingly. Fail below 35, Pass Class between 35 to 45, Second Class between 45 to 60, First Class between 60 to 70, Distinction if more than 70. (C)
4.	**Implementation of Control Structure (Part - II)**

	1. WAP to make a Simple Calculator using switch...case. (A) 2. WAP that prompts the user to input number of calls and calculate the monthly telephone bills as per the following rule: (B) Minimum Rs. 200 for up to 100 calls. Plus Rs. 0.60 per call for next 50 calls. Plus Rs. 0.50 per call for next 50 calls. Plus Rs. 0.40 per call for any call beyond 200 calls. 3. Three sides of a triangle are entered through the keyboard. WAP to check whether the triangle is isosceles, equilateral, scalene or right-angled triangle. (C)
5.	**Implementation of Control Structure (Part - III)** 1. WAP to print numbers between two given numbers which is divisible by 2 but not divisible by 3. (A) 2. WAP to find factorial of the given number. (A) 3. WAP to find whether the given number is prime or not without using function. (B) 4. WAP to print given number in reverse order. (C)
6.	**Demonstration of types of pattern using loop** 1. WAP to print following pattern using java. 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6 ****** * ** ** ** ** ****** 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 0 1 * *** ***** ******* ********* ********** ********* ******* ***** *** *
7.	**Demonstration of Function (Part-I)** 1. WAP to calculate simple interest using method. (A) 2. WAP to find maximum number from given three numbers using method. (A) 3. WAP to generate Fibonacci series of N given number using method. (B) 4. WAP to accept a number and check whether the number is prime or not. Use method name check (int n). The method returns 1, if the number is prime otherwise, it returns 0. (C)
8.	**Demonstration of Function (Part-II)** 1. WAP that calculates area of circle, triangle and square using method overloading. (A) 2. Write a method with following method header: public int gcd (int num1, int num2). (A) 3. Write a program that prompts the user to enter two integers and compute the gcd of two integers. [Note: The greatest common divisor (GCD) of two numbers is the largest number that divides them both.] (B) 4. WAP to find the factorial of given number using recursion. (C)
9.	**Implementation of single dimensional array** 1. WAP that create an array, take the size of array from the user, take the array member from the user and display it using loop. (A) 2. WAP to count number of even or odd number from an array of n number. (A) 3. WAP to accept n numbers in an array. Display the sum of all the numbers which are divisible by either 3 or 5. (B) 4. WAP to accept n numbers in an array. Now, enter a number and search whether the number is present or not in the list of array elements by using linear search. (C)
10.	**Implementation of multi dimensional array** 1. WAP to read values in two-dimensional array and print them in matrix form. (A) 2. WAP to print the score card of cricket match using the two dimensional array. Take all the input regarding score card from the user. (B) 3. WAP to read two matrices of size n X n, perform multiplication operation and store result in third matrix and print it. (C) 4. WAP to store numbers in 4 X 4 matrix in a two-dimensional array. Find the sum of the numbers of each row and the sum of numbers of each column of the matrix.(C)

11.	**Demonstration of the object and class (Part-I)** 1. Create a class "Student" that would contain enrolment No, name, and gender and marks as instance variables and count as static variable which stores the count of the objects; constructors and display(). Implement constructors to initialize instance variables. (A) 2. Create a class named Candidate with Candidate_ID, Candidate_Name, Candidate_Age, Candidate_Weight and Candidate_Height data members. Also create a method GetCandidateDetails() and DisplayCandidateDetails(). Create main method to demonstrate the Candidate class. (B) 3. Create a class named Bank_Account with Account_No, User_Name, Email, Account_Type and Account_Balance data members. Also create a method GetAccountDetails() and DisplayAccountDetails(). Create main method to demonstrate the Bank_Account class. (C)
12.	**Demonstration of the object and class (Part-II)** 1. WAP with following specifications: (A) Class Name: Employee Data Members: Employee_ID, Employee_Name, Designation, Age, Salary Member Functions: GetEmployeeDetails () and DisplayEmpolyeeDetails (). 2. Write a class program with following specifications: (B) Class Name: Student Data Members: Enrollment_No, Student_Name, Semester, CPI and SPI Member Functions: GetStudentDetails () and DisplayStudentDetails (). 3. WAP to create Circle class with area and perimeter function to find area and perimeter of circle. (C)
13.	**Demonstration of the object and class (Part-III)** 1. Define Time class with hour and minute as data member. Also define addition method to add two-time objects. (A) 2. Define class for Complex number with real and imaginary part. Describe its constructor, overload the constructors and instantiate its object. Also define addition method to add two Complex objects. (B) 3. Create a class which ask the user to enter a sentence, and it should display count of each vowel type in the sentence. The program should continue till user enters a word "quit". Display the total count of each vowel for all sentences. (C)
14.	**Implementation of built in function of string and stringbuffer class** 1. WAP to demonstrate the builtin function of String class (e.g. length(), charAt(), concat(), indexOf(), equals(), valueOf(), toString (), trim(), substring()) (A) 2. Java String Program to Print even length words. (B) 3. Java String Program to Insert a string into another string. (B) 4. Java String program to check whether a string is a Palindrome (C) 5. Java String Program to Set Characters to a String and get that character from that string. (C)
15.	**Demonstration of functionalities of math class** 1. WAP to demonstrate the builtin function of Math class (e.g. min(), max(), random(), pow(), sqrt(), round(), ceil(), floor(), abs()) (A) 2. WAP to print the largest number from the three given number using Math class function. (B) 3. WAP th check that weather the given number is Armstrong or not using Math class function. (C)
16.	**Demonstration of functionalities of Nested Class, Inner class, Wrapper class** 1. WAP to demonstrate concept of nested class. (A) 2. WAP to demonstrate annonymus inner class. (B) 3. WAP to demonstrate concepts of autoboxoing and unboxing using wrapper class. (C)
17.	**Implementation of of this and static keyword** 1. Write a program in Java to demonstrate use of this keyword. (A) 2. Check whether this can access the Static variables of the class or not.(A) 3. Write a java program static block which will be executed before main ( ) method in a class. Also demonstrate the static method. (B) 4. Create a class "Rectangle" that would contain length and width as an instance variable and count as a static variable. Define constructors [constructor overloading (default, parameterized and copy)] to initialize variables of objects. Define methods to find

	area and to display variables' value of objects which are created. [Note: define initializer block, static initializer block and the static variable and method. Also demonstrate the sequence of execution of initializer block and static initialize block] (C)
18.	**Implementation of types of inheritance** 1. WAP to demonstrate single inheritance, multilevel inheritance and hierarchical inheritance. (A) 2. Create a class named shape. In this class, we have three subclasses circle, triangle and square. WAP to display area of all three classes. (B) 3. WAP for implementing single inheritance which creates one class account_details for getting account information and another class interest for calculating and displaying total interest from the data inserted from account details. (C)
19.	**Implementation of function overriding, super and final keywords** 1. Demonstrate the Method overriding using example. (A) 2. Demonstrate the use of Super Keyword to access constructor, function and variable of immediate parent class from the subclass . (A) 3. Demonstrate the use of Final Keyword to prevent function overriding, variable modification and extraction of a class in sub class. (B) 4. Create a class named 'Member' having the following members: (C) 1 - Name 2 - Age 3 - Phone number 4 - Address 5 – Salary It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same along with specialization and department respectively.
20.	**Implementation of advanced concept of inheritance** 1. Design a class named MyPoint to represent a point with x- and y-coordinates. The class contains: The data fields x and y that represent the coordinates with getter methods. (A) o a no-arg constructor that creates a point (0, 0). o a constructor that constructs a point with specified coordinates. o a method named distance that returns the distance from this point to a specified point of the MyPoint type. o a method named distance that returns the distance from this point to another point with specified x- and y-coordinates. 2. Create a class named ThreeDPoint to model a point in a three-dimensional space. Let ThreeDPoint be derived from MyPoint with following additional features: (B) o A data fields named z that represents the z-coordinate. o A no-arg constructor that creates a point (0, 0, 0). o A constructor that constructs a point with three specified coordinates. o A get method that returns the z value. o Override the distance method to return the distance between two points in the three-dimensional space. 3. WAP that creates two points (0, 0, 0) and (10, 30, 25.5) and display the distance between the two points. (C)
21.	**Implementation of the use of abstract class** 1. An abstract Vegetable class has three subclasses named Potato, Brinjal and Tomato. Write a program that demonstrates how to establish this class hierarchy. Declare one instance variable of type String that indicates the color of a vegetable. Create and display instances of these objects. Override the toString() method of object to return a string with the name of vegetable and its color. (A) 2. Implement multilevel inheritance structure using class and interface. (B) 3. Implement hybrid inheritance structure using class and interface.(C)
22.	**Implementation of the use of interface** 1. Create interface EventListener with performEvent() method. Create MouseListener interface which inherits EventListener along with mouseClicked(), mousePressed(), mouseReleased(), mouseMoved(), mouseDragged() methods. Also create KeyListener interface which inherits EventListener along with keyPressed(), keyReleased() methods. WAP to create EventDemo class which implements MouseListener and KeyListener and demonstrate all the methods of the interfaces. (A)

	2. The Transport interface declares a deliver () method. The abstract class Animal is the super class of the Tiger, Camel, Deer and Donkey classes. The Transport interface is implemented by the Camel and Donkey classes. Write a test program that initialize an array of four Animal objects. If the object implements the Transport interface, the deliver () method is invoked. (B) 3. Declare a class called book having author_name as private data member. Extend book class to have two sub classes called book_publication & paper_publication. Each of these classes have private member called title. Write a program to show usage of dynamic method dispatch (dynamic polymorphism) to display book or paper publications of given author. Use command line arguments for inputting data. (C)
23.	**Demonstration of the types of access specifiers** 1. WAP to demonstrate the use of private, public, protected and default access modifiers using two package. (A) 2. Assume that there are two packages, student and exam. A student package contains Student class and the exam package contains Result class. Write a program that generates mark sheet for students. (B) 3. Define a class A in package apack. In class A, three variables are defined of access modifiers protected, private and public. Define class B in package bpack which extends A and write display method which accesses variables of class A. Define class C in package cpack which has one method display() in that create one object of class A and display its variables. Define class ProtectedDemo in package dpack in which write main () method. Create objects of class B and C and class display method for both these objects. (C)
24.	**Implementation of exception handling mechanism** 1. Demonstrate the built-in types of exception using example. (A) 2. Demonstrate the customize exception using throw keyword with example. (A) 3. WAP to accept N integer numbers from the command line. Raise and handle exceptions for following cases : (B) - when a number is –ve - when a number is evenly divisible by 10 - when a number is greater than 1000 and less than 2000 - when a number is greater than 7000 Skip the number if an exception is raised for it, otherwise add it to find total sum. 4. WAP to create Account class, which is representing a bank account where we can deposit and withdraw money. if we want to withdraw money which exceed our bank balance? We will not be allowed, create a customize exception to handle above situation and display proper error message. (C)
25.	**Implementation of File, FileReader and FileWriter class using java** 1. Write a java program to create a file to the specified location. (Use File Class) (A) 2. WAP to copy the content of one file to another file and console. (Use FileReader and File Writer Class) (A) 3. WAP to Merge the content of two files into single file. (B) 4. WAP to Copy the content of one file into multiple file. (C)
26.	**Demonstration of FileInputStream, FileOutputStream class** 1. WAP to demonstrate FileInputStream and FileOutputStream class. (A) 2. WAP to demonstrate BufferedReader and BufferedWriter class. (B) 3. WAP to demonstrate BufferedInputStream and BufferdOutputStream class. (C)

Useful Links
https://www.javatpoint.com/ https://www.geeksforgeeks.org/ https://www.tutorialspoint.com/ https://www.w3schools.com/