Java Programming Lab

BCA 3rd Year, 5th Semester

Section – C



Submitted To

Mr. Piyush Kumar Gupta

(Assistant Professor)

Submitted By:

Mohammad Abdus Khan

2018-310-052

**(Signature)**

**Jamia Hamdard**

**Department Of Computer Science & Engineering**

**School Of Engineering Sciences & Technology**

**New Delhi-110062**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Programs** | **Page No.** |
|  | WAP to print some statements like “Hello World!”. |  |
|  | WAP to calculate room area using multiple classes. |  |
|  | WAP to demonstrate the use of command line arguments. |  |
|  | WAP to explain the basic data types used in java. |  |
|  | WAP to explain the type casting in java. |  |
|  | WAP to demonstrate java expressions using different operators in java like relational, logical, bitwise operators etc. |  |
|  | WAP to demonstrate if-else, nested if-else, if-else ladder. |  |
|  | WAP to demonstrate switch statements. |  |
|  | WAP to demonstrate “? :” operator. |  |
|  | WAP to explain the working of while, for and do-while loops. |  |
|  | WAP to demonstrate different types of constructors in java like default and parameterized. |  |
|  | WAP to explain method overloading and constructor overloading. |  |
|  | WAP to explain static methods and static members. |  |
|  | WAP to demonstrate multilevel inheritance and super keyword. |  |
|  | WAP to demonstrate method overriding in hierarchical inheritance. |  |
|  | WAP to explain the working of final classes and also use final methods and final variables in your program. |  |
|  | WAP to explain the concept of finalization. |  |
|  | WAP to demonstrate the use of abstract methods and abstract classes. |  |
|  | WAP to demonstrate the use of access modifiers (Private, protected, public and default) |  |
|  | WA menu driven program to add, subtract and multiply two matrices. |  |
|  | WA menu driven program for selection sort, merge sort and quick sort. |  |
|  | WAP in which you have to make a table to store name of students, age, course, roll no and this table can be ordered in following ways:  1: Alphabetical order (A to Z or Z to A) of name  2: smallest to greatest or greatest to smallest order of age  3: Ascending or descending order of roll no  4: Bachelor to master or master to bachelor order of course |  |
|  | WAP to demonstrate the commonly used “String” and “StringBuffer” class methods. |  |
|  | WAP to demonstrate Wrapper class methods. |  |
|  | WAP to explain interfaces in java. |  |
|  | WAP to demonstrate multiple inheritances in java. |  |
|  | WAP to demonstrate package in java. |  |
|  | WAP to explain static import in java with package. |  |
|  | WAP to demonstrate threading in java using Thread class. |  |
|  | WAP to demonstrate threading in java using Runnable interface. |  |
|  | WAP to explain try and catch statements in java. |  |
|  | WAP to demonstrate exception handling using multiple catch statements and finally block. |  |
|  | WAP to demonstrate the use of “throw” and “throws” keyword in java. |  |
|  | WAP to demonstrate Applet with all the states used in it. |  |
|  | WAP to make graphic calculator. |  |
|  | Write two different programs to copy a file in another file by character by character and byte by byte methods. |  |
|  | Write a Java application to print Pascal’s triangle. |  |
|  | Write a program to compute the following series:  1 – x + x2/2! – x3/3! + x4/4! + ….. + (–/+ 1)nxn/n!  Where n and x is to be accepted by the user. |  |
|  | Define a class BankAccount with appropriate member variables and methods to deposit and withdraw amount. Refine BankAccount to SavingBankAccount and CurrentBankAccount using inheritance. Use them in main class to demonstrate dynamic method dispatch. |  |
|  | Create an abstract class Shape and derived classes Rectangle and Circle from Shape class. Implement abstract method of shape class in Rectangle and Circle class. Shape class contains: origin (x, y) as data member, display() and area() as abstract methods. Circle class contains: radius as data member. Rectangle class contains: length and width. (Use Inheritance, overloading and overriding concept) |  |
|  | Create a package named as MyPackage with a class named as Calculate. The class should contain three methods with the following specifications:   * Volume(): Accepts three double type arguments i.e. width, height, depth; calculate volume and return double type value. * Add(): Accepts two integer type values, adds them and returns the value. * Divide(): Accepts two integer type values, adds them and returns results.   Import this package into a file named as PackageDemo and call the above three methods. |  |
|  | Write a program that will count number of characters, words and lines in a file. The name should be passes as command line argument. |  |
|  | Write a multi-threaded Java program to print all numbers below 100,000 that are both prime and Fibonacci number (some examples are 2, 3, 5, 13, etc.). Design a thread that generates prime numbers below 100,000 and writes them into a pipe. Design another thread that generates Fibonacci numbers and writes them to another pipe. The main thread should read both the pipes to identify numbers common to both. |  |