1 Write a Java Program to define a class, define instance methods for setting and Retrieving values of instance variables and instantiate its object.

class Person { String name; int age;

Person(String name, int age) { this.name = name;

this.age = age;

}

String getName() { return name;

}

int getAge() {

return age;

}

}

class Test4 {

public static void main(String[] args) { Person[] people = new Person[3]; people[0] = new Person("John", 25); people[1] = new Person("Jane", 30); people[2] = new Person("Bob", 45);

for (int i = 0; i < people.length; i++) {

System.out.println("Person " + (i+1) + ": " +

people[i].getName() + " - " + people[i].getAge() + " years old");

}

}

2 Write a Java Program to define a class, define instance methods and overload them and use them for dynamic method invocation.

class Abc{

void run(){

System.out.println("Hiiii");

}

void run(String s){

System.out.println("Hello"+" "+s);

}

void run(int d){

System.out.println(d);

}

}

public class Simpl{

public static void main(String[] args) { Abc obj =new Abc();

obj.run(); obj.run("Utkarsh");

obj.run(21);

}

// method overloading

}

1. Write a Java Program to implement array of objects.

public class Studnt { private int id; private String name; private int age;

public Studnt(int id, String name, int age) { this.id = id; this.name = name; this.age = age; }

public int getId() {

return id; }

public String getName() {

return name; }

public int getAge() {

return age; }

public static void main(String[] args) { Studnt[] students = new Studnt[3]; students[0] = new Studnt(1, "John", 18); students[1] = new Studnt(2, "Jane", 19); students[2] = new Studnt(3, "Bob", 20); for (int i = 0; i < students.length; i++) {

System.out.println("Student ID: " + students[i].getId());

System.out.println("Student Name: " + students[i].getName());

System.out.println("Student Age: " + students[i].getAge());

}

}

}

Write a Java Program to implement Wrapper classes and their methods.

public class Test5{ public static void main(String args[]) { byte b=10; short s=20; int i=30; long l=40; float f=50.0F; double d=60.0D; char c='a'; boolean b2=true; Byte byteobj=b;

Short shortobj=s;

Integer intobj=i;

Long longobj=l;

Float floatobj=f;

Double doubleobj=d;

Character charobj=c;

Boolean boolobj=b2;

System.out.println("---Printing object values---");

System.out.println("Byte object: "+byteobj);

System.out.println("Short object: "+shortobj);

System.out.println("Integer object: "+intobj);

System.out.println("Long object: "+longobj);

System.out.println("Float object: "+floatobj);

System.out.println("Double object: "+doubleobj);

System.out.println("Character object: "+charobj); System.out.println("Boolean object: "+boolobj);

}

}

1. Write a Java Program to implement inheritance and demonstrate use of method overriding.

class Parent{ void Showme(){

System.out.println("Base Class method");

}

}

class child extends Parent{ void Showme(){

System.out.println("Child class method");

}

}

public class Test6 {

public static void main(String[] args) { child obj =new child(); obj.Showme();

} }

Write a Java Program to implement multilevel inheritance by applying various access controls to its data members and methods.

class Student{ private int rollno

String name;

Scanner sc= new Scanner(System.in); public void getrollno(){

System.out.println("Enter your rollno and name:"); rollno=sc.nextInt(); name=sc.nextLine();

}

void putrollno(){

System.out.println("Roll no:="+rollno);

System.out.println("Name="+name);

}

}

class Marks extends Student{ protected int m1,m2,m3; void getmarks(){ m1=sc.nextInt(); m2=sc.nextInt(); m3=sc.nextInt();

}

void putmarks(){

System.out.println("m1="+m1); System.out.println("m2="+m2); System.out.println("m3="+m3);

}

}

class Result extends Marks{

private float total; float avg; void compute\_display(){ total=m1+m2+m3;

System.out.println("Total Marks:"+total);

}

void average(){ avg=total/3;

System.out.println("Average Marks :"+avg);

}

} class Test7{

public static void main(String[] args) { Result r =new Result();

r.getrollno();

r.getmarks();

r.putrollno();

r.putmarks();

r.compute\_display();

r.average();

}

}

Write a Java program to demonstrate the use of implementing interfaces.

import java.util.Scanner; interface area{

public void dimensions();

public void area();

}

class Rectangle implements area{

int length,breadth,area; public void dimensions(){

Scanner sc = new Scanner(System.in); System.out.println("Enter Length:"); length=sc.nextInt();

System.out.println("Enter breath:"); breadth=sc.nextInt();

}

public void area(){

area=length\*breadth;

System.out.println("Area of Rectangl:"+area);

}

}

class Square implements area{ int length,area; public void dimensions(){

Scanner s = new Scanner(System.in); System.out.println("Enter Length:"); length=s.nextInt();

}

public void area(){

area=length\*length;

System.out.println("Area of the Square:"+area);

}

}

public class Test8 {

public static void main(String[] args) { Rectangle obj = new Rectangle(); obj.dimensions();

obj.area();

Square obj2 = new Square(); obj2.dimensions();

obj2.area();

}

}

Write a Java program to implement the concept of importing classes from user-defined packages and creating package.

package Mypackage;

public class Myclass { int a;

public void set\_value(int n){

a=n;

}

public void display\_value(){

System.out.println("the Value of a is : " + a);

}

}

import Mypackage.Myclass;; public class Pacakge {

public static void main(String[] args) { Myclass obj = new Myclass(); obj.set\_value(10);

obj.display\_value();

}

}

Write a program to implement the concept of threading by implementing Runnable Interface.

public class Test11 implements Runnable {

@Override public void run(){

for (int i=0;i<5;i++){

System.out.print(" "+i);

}

System.out.println();

}

public static void main(String[] args) {

Test11 obj = new Test11();

Thread t1 = new Thread(obj);

Thread t2 = new Thread(obj);

Thread t3 = new Thread(obj);

t1.start(); t2.start(); t3.start();

System.out.println("Hi");

}

}

Write a Java program to implement the concept of Exception Handling using predefined exception.

import java.util.Scanner;

public class Test12 {

public static void main(String[] args) { try{

Scanner sc = new Scanner(System.in); System.out.println("Enter Two Values:");

int n1=sc.nextInt(); int n2=sc.nextInt(); int n3=n1/n2;

System.out.println("DIVISION VALUE ="+n3);

}

catch(ArithmeticException Ae){

System.out.println("DONT ENTER ZERO FOR DENOMINATOR...");

}

catch(NumberFormatException Nfe){

System.out.println("PASS ONLY INTEGER

VALUES...");

}

finally{

System.out.println("I AM FROM FINALLY...");

}

}

}

Write a Java program to implement the concept ofException Handling by creating user defined exceptions.

Mypackage;

public class Myclass { int a;

public void set\_value(int n){

a=n;

}

public void display\_value(){

System.out.println("the Value of a is : " + a);

}

}

//second program import Mypackage.Myclass;; public class Pacakge {

public static void main(String[] args) { Myclass obj = new Myclass(); obj.set\_value(10);

obj.display\_value();

}

}