**Java Assignment No.4**

/\*

    Ques : Write a program to calculate area and volume of sphere using static

    variable and method create two static methods for area and volume

    calculation. (insert data from user).

 \*/

import java.util.Scanner;

public class Ques\_1 {

    private static final *double* PI = 3.14159265359;

    public static *double* calArea(*double* *radius*) {

        return 4 \* PI \* radius \* radius;

    }

    public static *double* calVolume(*double* *radius*) {

        return (4.0 / 3.0) \* PI \* radius \* radius \* radius;

    }

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

*Scanner* scanner = new Scanner(System.in);

        System.out.print("Enter the radius of the sphere: ");

*double* radius = scanner.nextDouble();

*double* area = calArea(radius);

*double* volume = calVolume(radius);

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

        System.out.println("Volume of the sphere: " + volume);

        scanner.close();

    }

}

/\*

    Ques : Display all your information (prn, name, age, address, class) on console

            without creating any object and writing any code in main method.

 \*/

public class Ques\_2 {

    private static *String* prn = "1234567890";

    private static *String* name = "John Doe";

    private static *int* age = 20;

    private static *String* address = "123 Main Street, City, Country";

    private static *String* className = "Class 10";

    static {

        System.out.println("PRN: " + prn);

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

        System.out.println("Age: " + age);

        System.out.println("Address: " + address);

        System.out.println("Class: " + className);

    }

    public static *void* main(*String* *args*[])

    {

    }

}

/\*

    Ques : Demonstrate how to use static inner class and non-static inner class to

            access static and non-static members of outer class.

 \*/

public class Ques\_3 {

    private static *int* staticOutVar = 10;

    private *int* nonstaticOutVar = 20;

    public static class StaticInnerClass {

        public *void* accessOuterStatic() {

            System.out.println("Accessing outer static variable from static inner class: " + staticOutVar);

        }

    }

    public class NonStaticInnerClass {

        public *void* accessOuterNonStatic() {

            System.out.println("Accessing outer static variable from non-static inner class: " + staticOutVar);

            System.out.println("Accessing outer non-static variable from non-static inner class: " + nonstaticOutVar);

        }

    }

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

*Ques\_3*.*StaticInnerClass* staticInnerObj = new Ques\_3.StaticInnerClass();

        staticInnerObj.accessOuterStatic();

*Ques\_3* outerObj = new Ques\_3();

*Ques\_3*.*NonStaticInnerClass* nonStaticInnerObj = outerObj.new NonStaticInnerClass();

        nonStaticInnerObj.accessOuterNonStatic();

    }

}

/\*

    Ques : Write a program using final variable to check speed limit exceeds or not

            on highway. If speed is greater than 100. Then generate alert message.

 \*/

import java.util.Scanner;

public class Ques\_4 {

    private final static *int* MAX\_SPEED\_LIMIT = 100;

    static *int* speed;

    static *Scanner* in = new Scanner(System.in);

    public *void* getSpeed()

    {

        System.out.print("Enter Speed : ");

        speed = in.nextInt();

    }

    public static *void* checkSpeed(*int* *speed*) {

        if (speed > MAX\_SPEED\_LIMIT) {

            System.out.println("Alert: Speed limit exceeded! Your speed is " + speed + " km/h, which exceeds the maximum limit of " + MAX\_SPEED\_LIMIT + " km/h.");

        } else {

            System.out.println("Your speed is within the limit.");

        }

    }

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

*Ques\_4* obj = new Ques\_4();

        obj.getSpeed();

        checkSpeed(speed);

    }

}

/\*

    Ques : Create an abstract class 'Bank' with an abstract method 'getBalance'.

            $100, $150 and $200 are deposited in banks A, B and C respectively.

            'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having

            a method named 'getBalance'. Call this method by creating an object of

            each of the three classes.

 \*/

// Abstract class Bank

abstract class Bank {

    abstract *int* getBalance();

}

// Subclass BankA

class BankA extends *Bank* {

    private *int* balance = 100;

*int* getBalance() {

        return balance;

    }

}

// Subclass BankB

class BankB extends *Bank* {

    private *int* balance = 150;

*int* getBalance() {

        return balance;

    }

}

// Subclass BankC

class BankC extends *Bank* {

    private *int* balance = 200;

*int* getBalance() {

        return balance;

    }

}

class Ques\_5{

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

*BankA* bankA = new BankA();

*BankB* bankB = new BankB();

*BankC* bankC = new BankC();

        System.out.println("Balance in BankA: $" + bankA.getBalance());

        System.out.println("Balance in BankB: $" + bankB.getBalance());

        System.out.println("Balance in BankC: $" + bankC.getBalance());

    }

}

/\*

    Ques : An abstract class has a construtor which prints "This is constructor of

        abstract class", an abstract method named 'a method' and a non-abstract

        method which prints "This is a normal method of abstract class". A class

        'SubClass' inherits the abstract class and has a method named 'a method'

        which prints "This is abstract method". Now create an object of

        'SubClass' and call the abstract method and the non-abstract method.

\*/

// Abstract class

abstract class AbstractClass {

    AbstractClass() {

        System.out.println("This is constructor of abstract class");

    }

    abstract *void* a\_method();

*void* normalMethod() {

        System.out.println("This is a normal method of abstract class");

    }

}

// Subclass

class SubClass extends *AbstractClass* {

*void* a\_method() {

        System.out.println("This is abstract method");

    }

}

public class Ques\_6 {

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

*SubClass* obj = new SubClass();

        obj.a\_method();

        obj.normalMethod();

    }

}

/\*

    Ques : We have to calculate the area of a rectangle, a square and a circle. Create

            an abstract class 'Shape' with three abstract methods namely

            'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea'

            taking one parameter each. The parameters of 'RectangleArea' are its

            length and breadth, that of 'SquareArea' is its side and that of 'CircleArea'

            is its radius. Now create another class 'Area' containing all the three

            methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the

            area of rectangle, square and circle respectively. Create an object of class

            'Area' and call all the three methods.

 \*/

// Abstract class Shape

abstract class Shape {

    abstract *double* RectangleArea(*double* *length*, *double* *breadth*);

    abstract *double* SquareArea(*double* *side*);

    abstract *double* CircleArea(*double* *radius*);

}

class Area extends *Shape* {

*double* RectangleArea(*double* *length*, *double* *breadth*) {

        return length \* breadth;

    }

*double* SquareArea(*double* *side*) {

        return side \* side;

    }

    @*Override*

*double* CircleArea(*double* *radius*) {

        return Math.PI \* radius \* radius;

    }

}

public class Ques\_7 {

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

*Area* area = new Area();

*double* rectangleArea = area.RectangleArea(5, 10);

        System.out.println("Area of rectangle: " + rectangleArea);

*double* squareArea = area.SquareArea(7);

        System.out.println("Area of square: " + squareArea);

*double* circleArea = area.CircleArea(3);

        System.out.println("Area of circle: " + circleArea);

    }

}

/\*

    Ques : Define a package named 'useful' with a class name 'Useme' having

            following methods:

            1. area → To calculate area of given shape.

            2. percentage → to calculate percentage given total marks and marks

            obtained.

            Develop a program to import above package and use both methods.

 \*/

import useful.Useme;

public class Main {

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

*Useme* useme = new Useme();

*double* rectangleArea = useme.area(5, 10);

        System.out.println("Area of rectangle: " + rectangleArea);

*double* totalMarks = 500;

*double* obtainedMarks = 450;

*double* percentage = useme.percentage(totalMarks, obtainedMarks);

        System.out.println("Percentage obtained: " + percentage + "%");

    }

}

package useful;

public class Useme {

    public *double* area(*double* *length*, *double* *breadth*) {

        return length \* breadth;

    }

    public *double* percentage(*double* *totalMarks*, *double* *obtainedMarks*) {

        return (obtainedMarks / totalMarks) \* 100;

    }

}