Java Assignment-4

22610011-Anuja Suntnur

1. 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 Sphere {
   static double pi = 3.14159265359;

   static double calculateArea(double radius) {
      return 4 * pi * radius * radius;
   }

   static double calculateVolume(double radius) {
      return (4 * pi * radius * radius * radius) / 3;
   }

   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 = calculateArea(radius);
      double volume = calculateVolume(radius);
      System.out.println("Area of the sphere: " + area);
      System.out.println("Volume of the sphere: " + volume);
```

```
        scanner.close();
    }
}
```

Output:

```
Enter the radius of the sphere: 4
Area of the sphere: 201.06192982976
Volume of the sphere: 268.0825731063467
```

2. Display all your information (prn, name, age, address, class) on console without creating any object and writing any code in main method.

```
class Student {
   static String prn = "22610011";
   static String name = "Anuja Suntnur";
   static int age = 19;
   static String address = "Solapur";
   static String batch = "S2";

   static {
      System.out.println("PRN: " + prn);
      System.out.println("Name: " + name);
      System.out.println("Age: " + age);
      System.out.println("Address: " + address);
      System.out.println("Batch: " + batch);
   }
}
class StudentInfo {
   public static void main(String[] args){
}}
```

Output:

```
PRN:  22610011
Name:  Anuja Suntnur
Age:  19
Address:  Solapur
Batch:  S2
```

3. Demonstrate how to use static inner class and non-static inner class to access static and non-static members of outer class.

```
public class Outer {
   static int outerStaticVariable = 10;
   int outerInstanceVariable = 20;

   static class StaticInner {
      void display() {
         System.out.println("Outer static variable: " +
outerStaticVariable);
         // Cannot access outerInstanceVariable here
      }
   }

   class NonStaticInner {
      void display() {
         System.out.println("Outer static variable: " +
outerStaticVariable);
```

```
        System.out.println("Outer instance variable: " + outerInstanceVariable);
    }
  }

  public static void main(String[] args) {
    StaticInner staticInner = new StaticInner();
    staticInner.display();

    Outer outer = new Outer();
    NonStaticInner nonStaticInner = outer.new NonStaticInner();
    nonStaticInner.display();
  }
}
```

Output:

```
Outer static variable: 10
Outer static variable: 10
Outer instance variable: 20
```

4. 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 SpeedChecker {
   public static final int SPEED_LIMIT = 100;

   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter the speed of the car: ");
      int speed = scanner.nextInt();
      if (speed > SPEED_LIMIT) {
         System.out.println("Speed limit exceeded! Alert message generated.");
      } else {
         System.out.println("Within speed limit.");
      }
      scanner.close();
   }
```

}

Output:

```
Enter the speed of the car: 120
Speed limit exceeded! Alert message generated.
```

5. 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 int getBalance();
}

class BankA extends Bank {
   @Override
   int getBalance() {
      return 100;
   }
}

class BankB extends Bank {
   @Override
   int getBalance() {
      return 150;
   }
```

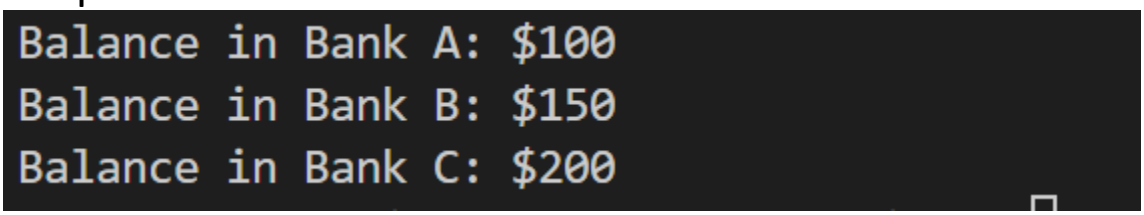
```
}

class BankC extends Bank {
    @Override
    int getBalance() {
        return 200;
    }
}

public class BankTest {
    public static void main(String[] args) {
        BankA bankA = new BankA();
        BankB bankB = new BankB();
        BankC bankC = new BankC();

        System.out.println("Balance in Bank A: $" +
bankA.getBalance());
        System.out.println("Balance in Bank B: $" +
bankB.getBalance());
        System.out.println("Balance in Bank C: $" +
bankC.getBalance());
    }
}
```

Output:



6. 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 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");
   }
}

class SubClass extends AbstractClass {
   @Override
```

```
    void a_method() {
        System.out.println("This is abstract method");
    }
}

public class AbstractExample {
    public static void main(String[] args) {
        SubClass subClass = new SubClass();
        subClass.a_method();
        subClass.normalMethod();
    }
}
```

Output:

```
This is constructor of abstract class
This is abstract method
This is a normal method of abstract class
```

7. 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 double RectangleArea(double length, double breadth);

   abstract double SquareArea(double side);

   abstract double CircleArea(double radius);
}

class Area extends Shape {
   @Override
   double RectangleArea(double length, double breadth) {
      return length * breadth;
```

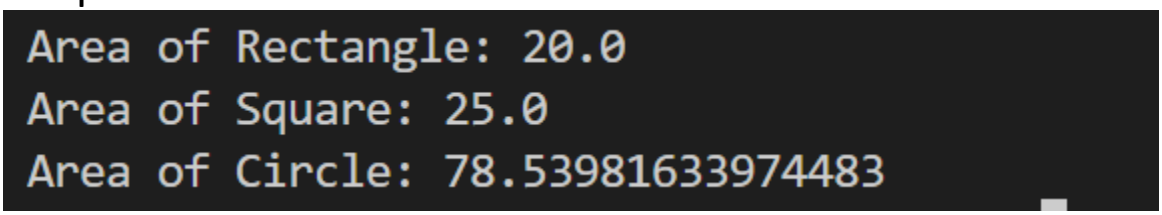
```
    }

    @Override
    double SquareArea(double side) {
        return side * side;
    }

    @Override
    double CircleArea(double radius) {
        return Math.PI * radius * radius;
    }
}

public class AreaTest {
    public static void main(String[] args) {
        Area area = new Area();
        System.out.println("Area of Rectangle: " +
area.RectangleArea(5, 4));
        System.out.println("Area of Square: " +
area.SquareArea(5));
        System.out.println("Area of Circle: " + area.CircleArea(5));
    }
}
```

Output:



8. 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.

```
package useful;

public class Useme {
   public double area(double length, double breadth) {
      return length * breadth;
   }

   public double percentage(double totalMarks, double marksObtained) {
      return (marksObtained / totalMarks) * 100;
   }
}

import useful.Useme;

public class UseTest {
   public static void main(String[] args) {
```

```
        Useme useme = new Useme();
        double rectangleArea = useme.area(5, 4);
        System.out.println("Area of Rectangle: " + rectangleArea);

        double percentage = useme.percentage(100, 85);
        System.out.println("Percentage obtained: " + percentage);
    }
}
```

Output: