**Assignment 3**

* Explain polymorphism.

It means one thing has many behaviors

* What is overloading?

They have same method name, they may have different arguments and return type

* What is overriding?

They have same method name , paramter and return type. just add more information in the method.

* What does the final mean in this method: public void doSomething(**final** Car aCar){}

It means Car class is a immuable class.

* Suppose in question 4, the Car class has a method setColor(Color color){…}, inside doSomething method, Can we call aCar.setColor(red);

No

* Can we declare a static variable inside a method?

No

* What is the difference between interface and abstract class?

Interface has only has abstract method and default method, the abstract class has non-abstract and abstract method. Interface has final variable, and abstract class has final variable and non-final varable.

* Can an abstract class be defined without any abstract methods?

Yes

* Since there is no way to create an object of abstract class, what’s the point of constructors of abstract class?
* What is a native method?

A native method is a method that is implemented in a language other than Java

* What is marker interface?

If a interface does not contain any method or fields and by implementing that interface if our objects will get some ability such type of interfaces are called as Marker interfaces

* Why to override equals and hashCode methods?

if the class work with hashmap, we do not override this two method we will get dupcatle value.

* What’s the difference beween int and Integer?

Int is primatiy type and Integer is wrap class.

* What is serialization?

To *serialize* an object means to convert its state to a byte stream so that the byte stream can be reverted back into a copy of the object.

* Create List and Map. List A contains 1,2,3,4,10(integer) . Map B contains ("a","1") ("b","2") ("c","10") (key = string, value = string)

list<Integer> list = Arrays.asList(1,2,3,4,10);

Map<String,String> map = new HashMap<>();

map.put("a","1");

map.put("b","2");

map.put("c","10")

for(Integer i:list){

if( !map.contains(i)){

list.add(i);

}

}

Question: get a list which contains all the elements in list A, but not in map B.

* Implement a group of classes that have common behavior/state as Shape. Create Circle, Rectangle and Square for now as later on we may need more shapes. They should have the ability to calculate the area. They should be able to compare using area. Please write a program to demonstrate the classes and comparison. You can use either abstract or interface. Comparator or Comparable interface.

public abstract class Shape {

public abstract double area();

public abstract double perimeter();

}

Implement a Circle, Triangle, and Rectangle class which extend the class Shape.

Ex: public class Circle extends Shape ... etc

public abstract class Shape {

public abstract double area();

public abstract double perimeter();

}

public class Rectangle extends Shape {

private final double width, length; //sides

public Rectangle() {

this(1,1);

}

public Rectangle(double width, double length) {

this.width = width;

this.length = length;

}

@Override

public double area() {

// A = w \* l

return width \* length;

}

@Override

public double perimeter() {

// P = 2(w + l)

return 2 \* (width + length);

}

}

public class Circle extends Shape {

private final double radius;

final double pi = Math.PI;

public Circle() {

this(1);

}

public Circle(double radius) {

this.radius = radius;

}

@Override

public double area() {

// A = π r^2

return pi \* Math.pow(radius, 2);

}

public double perimeter() {

// P = 2πr

return 2 \* pi \* radius;

}

}

public class Triangle extends Shape {

private final double a, b, c; // sides

public Triangle() {

this(1,1,1);

}

public Triangle(double a, double b, double c) {

this.a = a;

this.b = b;

this.c = c;

}

@Override

public double area() {

// Heron's formula:

// A = SquareRoot(s \* (s - a) \* (s - b) \* (s - c))

// where s = (a + b + c) / 2, or 1/2 of the perimeter of the triangle

double s = (a + b + c) / 2;

return Math.sqrt(s \* (s - a) \* (s - b) \* (s - c));

}

@Override

public double perimeter() {

// P = a + b + c

return a + b + c;

}

}