**Assignment 3**

1. Explain polymorphism.

A generic can have multiple kinds of implementation

1. What is overloading?

Method with different parameter

1. What is overriding?

Runtime polymorphism

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

aCar can’t be changed.

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

Yes

1. Can we declare a static variable inside a method?

No

1. What is the difference between interface and abstract class?

Class inherit multiple interface but only one abstract class.

Abstract class can have real method but interface is not/

1. Can an abstract class be defined without any abstract methods?

Yes

1. Since there is no way to create an object of abstract class, what’s the point of constructors of abstract class?

Initiate field of abstract class

1. What is a native method?

Using C/C++

1. What is marker interface?

Mark the method

1. Why to override equals and hashCode methods?

Reduce hash conflict, default does no work

1. What’s the difference beween int and Integer?

Integer is wrapper of int.

1. What is serialization?

Convert to byte stream.

1. 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)

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

C = new list();

For (int b:B.valueSet()){

If (!A.contains(b)){

c.add(b);

}

}

1. 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.

interface Shape {

int area();

}

class Rectangle implements Shape(){

int h;

int w;

Rectangle(int h, int w) {

this.h = h;

this.w = w;

}

int area() {

return h\*w;

}

}

class Square implements Shape {

int w;

Square(int w) {

this.w = w;

}

Int area(){

Return w\*w;}

}

class Circle implements Shape {

int r;

Circle(int r) {

this.r = r;

}

int area() {

return r \* r \* 3.14;

}

}

@Override

public int compareTo(Square s) {

if(this.area() == s.area()){

return 0;

}

if(this.area() < s.area()){

return -1;

}else{

return 1;

}

}