**Assignment 3**

1. Explain polymorphism.

Polymorphism is the ability of an object to take on many forms. It occurs when we have many classes that are related to each other by inheritance.

1. What is overloading?

Generally there are method overloading and operator overloading like in C++. But Java has only method overloading and it means that there are multiple functions with the same name but different parameters.

1. What is overriding?

After inheriting a superclass, if the class rewrite some methods which have the same name and parameters as in the superclass, this is co called overriding.

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

Java pass the aCar object by value which means any modification in the function call won’t influence original aCar object.

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

Yes if red is an instance of Color or there is a methods called “setColor” which has single input argument and the argument data type is same as the red object.

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

No. A static variable belongs to the class and it will be loaded into memory along with the class. Variables in methods means they should be destroyed after the method execution so it make no sense to create a static variable inside a method. But actually you can declare and there would be a compile error waiting for you.

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

Interface support multiple implementations while abstract class does not support multiple inheritance.

Abstract class can have data members while interface cannot.

Abstract class has constructor while interface do not.

Interface only has incomplete members while abstract class can have complete and incomplete members.

Interface can only have public access modifier while abstract class can have any access modifier.

No member of interface can be static while complete members can be static in abstract class.

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?

Main purpose of a constructor in an abstract class is for initializing fields. If we want it to initialize fields in certain way every time, then you can call “super()” with required parameters if you need to use the constructor in the abstract parent class.

1. What is a native method?

Native method allows you to call a compiled dynamically loaded library with arbitrary assembly code from java and get the results back to Java.

1. What is marker interface?

A marker interface is an [interface](https://www.baeldung.com/java-interfaces) that has no methods or constants inside it. It provides run-time type information about objects, so the compiler and JVM have additional information about the object.

1. Why to override equals and hashCode methods?

Collections such as HashMap and HashSet use a hashcode value of an object to determine how it should be stored inside a collection, and the hashcode is used again in order to locate the object in its collection.

Hashing retrieval is a two-step process:

1. Using the hashcode() to find the right bucket.
2. Using equals() to search the bucket for the right element

By defining equals() and hashCode() consistently, you can improve the usability of your classes as keys in hash-based collections.

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

“Int” is a primitive data type and “Integer” is a class data type. This Integer wrapper class can wrap a “int” into an object.

1. What is serialization?

Object serialization means an object can be represented as a sequence of bytes that includes the object's data as well as information about the object's type and the types of data stored in the object. After the serialization, data can to stored and retrieved back by deserialization.

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.

1. Implement a group of classes that have common behavior/state as Shape. Cre"ate 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.