Assignment 3

1. Explain polymorphism.
   Polymorphism is the ability of an object to take on different forms. In java, polymorphism refers to the ability of a class to provide different implementations of a method, depending on the type of object that is passed to the method.

2. What is overloading?
   Overloading happens in the same class. When method name is the same but with different parameters. E.g. constructor overloading

3. What is overriding?
   Overriding enable subclass to provide different implementation for a method in parent class.

4. What does the final mean in this method: public void doSomething(**final** Car aCar){}
   The reference cannot be change. aCar cannot refer to other objects. It can only refer to the object pass to the method.

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

6. Can we declare a static variable inside a method?
   No

7. What is the difference between interface and abstract class?
   Abstract class can have both abstract and not-abstract method, and non-static fields. But inter face can have public abstract methods and public static fields. Class can extend one abstract but implements multiple interfaces. Abstract class can implement interface or extend class. Interface can only extend interface.

8. Can an abstract class be defined without any abstract methods?
   No

9. Since there is no way to create an object of abstract class, what's the point of constructors of abstract class?
   - The main purpose of the constructor is to initialize the newly created object. In abstract lcass, we have an instance variable, abstract methods and non-abstract methods. We need to initialize the non-abstract methods and instance variables, therefore abstract classes have a constructor.
   - Also, even if we don't provide any constructor the compiler will ad default constructor in an abstract class.
   - An abstract class can be inherited by any number of sub-classes, thus functionality of constructor present in abstract class can be used by them

- The constructor inside the abstract class can be only called during constructor chaining i.e. when we create an instance of sub-classes. This also one of the reasons abstract class can have a constructor.

10. What is a native method?
    It is used to merge the efficiency and functions of C and C++ in the Java program. When the Java compiler did not implement or support some function, then, in that case, to increase the performance of the java application, the native methods are used.

11. What is marker interface?
    A marker interface is an interface that has no methods or constants inside it.

12. Why to override equals and hashCode methods?
    We know that two object are considered equal only if their references point to the same object, and unless we override equals and hashCode methods, the class object will not behave properly on hash-based collections like HashMap, HashSet, and Hashtable. This is because hash-based collections are organized like a sequence of buckets, and the hash code value of an object is used to determine the bucket where the object would be stored, and the same hash code is used again to find the object's position in the bucket.

13. What's the difference beween int and Integer?
    Int is a primitive data type has got less flexibility. We can only store the binary value of an integer. Since the integer is a wrapper class for int data type, it gives us more flexibility in storing, converting and manipulating an int data. integer is a class and thus it can call various in-built methods defined in the class.

14. What is serialization?
    Serialization is the process of converting an object into a stream of bytes to store the object or transmit it to memory, a database, or a file. Its main purpose is to save the state of an object in order to be able to recreate.

15. 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.

```
List<Integer> listA = Arrays.asList(1,2,3,4,10);
Map<String, String> mapB = new HashMap<>();
mapB.put("a", "1");
mapB.put("b", "2");
mapB.put("c", "10");
List<Integer> list = new LinkedList<>();
for(Integer elem : listA){
    if(!mapB.containsValue(elem.toString())){
        list.add(elem);
    }
}
```

16. 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.

    See Assignment3.java