Assignment 3

1. Explain polymorphism.
   Polymorphism in Java is the ability of an object to take many forms.
   Override and Overload

2. What is overloading?
   Method names are same, but the arguments are different. Return type can be anything, it doesn't matter.

3. What is overriding?
   Overriding is done so that a child class can give its own implementation to a method which is already provided by the parent class.

4. What does the final mean in this method: public void doSomething(**final** Car aCar){}
It means aCar variable can not be re-assigned the doSomething 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, we can call it.

6. Can we declare a static variable inside a method?
   No, we can not.

7. What is the difference between interface and abstract class?
   Abstract classes can have constants, members, method stubs (methods without a body) and defined methods, whereas interfaces can only have constants and methods stubs.

Interface	Abstract class
Interface support multiple implementations.	Abstract class does not support multiple inheritance.
Interface does not contain Data Member	Abstract class contains Data Member
Interface does not contain Constructors	Abstract class contains Constructors
An interface Contains only incomplete member (signature of member)	An abstract class Contains both incomplete (abstract) and complete member
An interface cannot have access modifiers by default everything is assumed as public	An abstract class can contain access modifiers for the subs, functions, properties
Member of interface can not be Static	Only Complete Member of abstract class can be Static

8. Can an abstract class be defined without any abstract methods?
   Yes we can have an abstract class without Abstract Methods as both are independent concepts.

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

Abstract classes require constructors to enforce a design contract using which objects can be initialized. Constructor can be used to setting up an attribute of the base class. This attribute will be available in subclass by inheritance. Before using this attribute, using the constructor it can be initialized.

10. What is a native method?

Native methods are Java methods that start in a language other than Java. Native methods can access system-specific functions and APIs that are not available directly in Java.

11. What is marker interface?

A marker interface is an interface 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.

12. Why to override equals and hashCode methods?

Because failure to override them so will result in a violation of the general contract for Object.hashCode(), which will prevent your class from functioning properly in conjunction with all hash-based collections, including HashMap, HashSet, and Hashtable.

13. What's the difference beween int and Integer?

int is a primitive type, and Integer is an object.

14. What is serialization?

Serialization is the process of turning an object in memory into a stream of bytes so you can do stuff like store it on disk or send it over the network.

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.

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.