**Assignment 3**

1. **Explain polymorphism.**Polymorphism normally happens during inheritance. Each subclass is a unique form of the parent class. For the methods that are in the parent class, subclasses can override them in different ways.
2. **What is overloading?**Overload is to enable a different parameter under the same method in the same class.
3. **What is overriding?**Overrideis the implementation in the child class that provide the specific implementation of a method that provided by one of its parent classes.
4. **What does the final mean in this method: public void doSomething(final Car aCar){}**The final keyword in this method make the aCar cannot be reassigned to a new object reference which mean that the passing object in this method is cannot be assigned to a new object, in this case a new Car.
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. Since the setter is inside of the class which the final keyword won’t affect at all.
6. **Can we declare a static variable inside a method?**No, static variable must be declared directly under the class but not in local area.
7. **What is the difference between interface and abstract class?**An **abstract** class permits you to make functionality that child classes can implement or override whereas an interface only permits you to state functionality but not to implement it. A class can extend only one abstract class while can implement multiple interfaces. **Abstract** class normally has a mixture of abstract and non-abstract methods. These methods can be public/protected/private. In contrast, **Interface** normally has only abstract and public methods. **Interface** has started supporting default and static methods since Java 8, which are allowed to be implemented in the interface. **Abstract** class does not support multiple inheritance whereas **Interface** support that. **Abstract** class allow final, static, and non-static variables whereas **Interface** only allow static and final variables. **Abstract** class can implement an interface whereas **Interface** cannot extend an abstract class.
8. **Can an abstract class be defined without any abstract methods?**

Yes, we can declare an abstract class with no abstract methods in Java.

1. **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 class, we have an instance variable, abstract methods, and non-abstract methods. We need to initialize the 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 add 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.
2. **What is a native method?**A native method is a Java method whose implementation is also written in another programming language such as C/C++.  
   We can use it to implement an interface with system calls or libraires written in other programming languages. It can access system or hardware resources that are only reachable from the other language. It can integrate already existing legacy code written in C/C++ into a Java application. It can call a compiled dynamically loaded library with arbitrary code from Java.
3. **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, so the compiler and JVM have additional information about the object.
4. **Why to override equals and hashCode methods?**Because if we want to store our custom object into a hash-based collections such as HashMap, HashTable, etc. It will call the equals() and hashCode() to compare the object being inserted and the hashing value for the target bucket. If we do not override these two methods, the collections will not be working fine.
5. **What’s the difference between int and Integer?**A int a primitive data type in Java which store 32 bits signed 2’s compliment integer. Whereas Integer is a wrapper class which wraps a primitive type int into an object.
6. **What is serialization?**Serialization is the process of turning an object in memory into a stream of bytes so we can do the very base level of operation on them, such as store them into the disk, send them over the internet socket stream.
7. **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.**<https://github.com/buyichen/antra_java_batch_5_6/tree/main/src/main/java/day4/Assignment3/Question15>

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.**<https://github.com/buyichen/antra_java_batch_5_6/tree/main/src/main/java/day4/Assignment3/Question16>