Java Day 2

Basic Concepts

1. What is the purpose of Garbage Collection in Java?

The purpose of garbage collection is to identify and discard objects that are no longer needed by a program so that their resources can be reclaimed and reused.

A Java object is subject to garbage collection when it becomes unreachable to the program in which it is used.

2. What is synchronization?

With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources.

Without synchonization, it is possible for one thread to modify a shared variable while another thread is in the process of using or updating same shared variable. This usually leads to significant errors.

3. How to create a Thread?

Thread can be implemented by using runnable interface or by inheriting from the Thread class.

The former is more advantageous, because it enables multiple inheritance.

4. What is pass by reference vs. pass by value?

Pass By Reference means the passing the address itself rather than passing the value.

Passby Value means passing a copy of the value to be passed.

5. Difference between HashMap and Map?

Map is Interface and Hashmap is class that implements that.

6. What is the difference between HashMap and HashTable?

The HashMap class is roughly equivalent to Hashtable, except that it is unsynchronized and permits nulls.

HashMap allows null values as key and value whereas Hashtable doesnt allow.

HashMap does not guarantee that the order of the map will remain constant over time.

HashMap is unsynchronized and Hashtable is synchronized.

7. Difference between Vector and ArrayList?

Vector is synchronized whereas arraylist is not.

8. Difference between Swing and AWT?

AWT are heavy-weight componenets.

Swing components are light-weight components.

Hence swing works faster than AWT.

9. What is the difference between a constructor and a method?

A constructor is a member function of a class that is used to create objects of that class. It has the same name as the class itself, has no return type, and is invoked using the new operator.

A method is an ordinary member function of a class. It has its own name, a return type (which may be void), and is invoked using the dot operator.

10. What is an Iterator?

Some of the collection classes provide traversal of their contents via a java.util.Iterator interface.

This interface allows you to walk through a collection of objects, operating on each object in turn.

Remember when using Iterators that they contain a snapshot of the collection at the time the Iterator was obtained; generally it is not advisable to modify the collection itself while traversing an Iterator.

11. What are access modifiers?

The following are the main access modifiers in Java :

public : Public class is visible in other packages, field is visible everywhere (class must be public too)

private : Private variables or methods may be used only by an instance of the same class that declares the variable or method, A private feature may only be accessed by the class that owns the feature.

protected : Is available to all classes in the same package and also available to all subclasses of the class that owns the protected feature.This access is provided even to subclasses that reside in a different package from the class that owns the protected feature.

default :What you get by default ie, without any access modifier (ie, public private or protected).It means that it is visible to all within a particular package.

13. What is an abstract class?

Abstract class must be extended/subclassed (to be useful). It serves as a template. A class that is abstract may not be instantiated (ie, you may not call its constructor), abstract class may contain static data. Any class with an abstract method is automatically abstract itself, and must be declared as such.

A class may be declared abstract even if it has no abstract methods. This prevents it from being instantiated.

14. What is static in java?

Static means one per class, not one for each object no matter how many instance of a class might exist.

This means that you can use them without creating an instance of a class.

Static methods are implicitly final, because overriding is done based on the type of the object, and static methods are attached to a class, not an object.

A static method in a superclass can be shadowed by another static method in a subclass, as long as the original method was not declared final.

However, you can't override a static method with a nonstatic method.

In other words, you can't change a static method into an instance method in a subclass.

15. What is final in java?

A final class can't be extended ie., final class may not be subclassed.

A final method can't be overridden when its class is inherited.

You can't change value of a final variable (is a constant).