

## Unit-II JAVA

### 2 Mark Question and Answer

#### 1) Mention some of the separators used in java programming? (AU NOV-DEC2012)

( )	Encloses arguments in method definitions and calling; adjusts precedence in arithmetic expressions; surrounds cast types and delimits test expressions in flow control statements
{ }	defines blocks of code and automatically initializes arrays
[ ]	declares array types and dereferences array values
;	terminates statements
,	separates successive identifiers in variable declarations; chains statements in the test, expression of a for loop
.	Selects a field or method from an object; separates package names from sub-package and class names
:	Used after loop labels

#### 2) What is mean by abstract base class?(AU NOV-DEC2012)

An abstract class is, conceptually, a class that cannot be instantiated and is usually implemented as a class that has one or more pure virtual (abstract) functions.

A pure virtual function is one which must be overridden by any concrete (i.e., non-abstract) derived class. This is indicated in the declaration with the syntax "= 0" in the member function's declaration.

#### 3) How dynamic initialization variables are achieved in java?(AU NOV-DEC-2012)

Initialization is the process of providing value to a variable at declaration time. A variable is initialized once in its life time. Any attempt of setting a variable's value after its declaration is called assignment. To use a local variable you have to either initialize or assign it before the variable is first used. But for class members, the compulsion is not so strict. If you don't initialize them then compiler takes care of the initialization process and set class members to default values.

#### 4) What is an object? Give example (AU NOV-DEC2014)

Object - Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behaviors -wagging, barking, eating. An object is an instance of a class.

#### 5) Define an access specifier ?give example.(AU NOV-DEC2014)

Java provides a number of access modifiers to set access levels for classes, variables, methods and constructors. The four access levels are:

- Visible to the package. The default. No modifiers are needed.
- Visible to the class only (private).
- Visible to the world (public).

- Visible to the package and all subclasses (protected)

### 6) What is polymorphism? (AU NOV DEC 2012)

Polymorphism is a feature that allows one interface to be used for a general class of actions. It is taking more than one form.

### 7) What are the kinds of variables in Java? (AU NOV/DEC 2011)

Java has three kinds of variables namely,

- a. Instance variables
- b. Local variables
- c. Class variables

### 8) Describe output streams and input streams in Java.( AU NOV DEC 2012)

I/O in Java is built on streams. Input streams read data. Output streams write data. Different fundamental stream classes such as `java.io.FileInputStream` and `sun.net.TelnetOutputStream` read and write particular sources of data.

### 9) Enumerate two situations in which static methods used?(AU MAY-JUNE-2014)

Factory methods

- That return a particular instance like in a Singleton: `java.lang.Runtime.getRuntime()`
- That return a subclass of a super class as you mentioned: `java.util.Calendar.getInstance()`
- That perform type conversion: `String.valueOf(int)`.

Utility methods

- In a class that also has instance methods : `java.lang.String.format()`
- In utility classes that are not meant to be instantiated at all: `java.util.Arrays`, `java.lang.Math`, `java.util.Collection`.

### 10) Mention the purpose of the keyword final?(AU MAY-JUNE2013)

Final in java is very important keyword and can be applied to class, method, and variables in Java. A variable can be declared as **final**. Doing so prevents its contents from being modified. This means that you must initialize a **final** variable when it is declared

### 11) What is the difference between constructor and method?(AU MAY-JUNE2013)

Constructors can't be called directly; they are called implicitly when the new keyword creates an object. Methods can be called directly on an object that has already been created with new.

constructors create and initialize objects that don't exist yet, while methods perform operations on objects that already exist.

### 12) How to define an interface?(AU MAY-JUNE2013)

Declaring Interfaces:

The interface keyword is used to declare an interface. Here is a simple example to declare an interface:

Example:

```
/* File name : NameOfInterface.java */
```

```
import java.lang.*;
//Any number of import statements

public interface NameOfInterface
{
    //Any number of final, static fields
    //Any number of abstract method declarations\
}
```

### **13) What is meant by Object Oriented Programming?**

OOP is a method of programming in which programs are organised as cooperative collections of objects. Each object is an instance of a class and each class belong to a hierarchy

### **14) What is a Class?**

Class is a template for a set of objects that share a common structure and a common behaviour.

### **15) What is meant by abstraction?**

Abstraction defines the essential characteristics of an object that distinguish it from all other kinds of objects. Abstraction provides crisply-defined conceptual boundaries relative to the perspective of the viewer. Its the process of focussing on the essential characteristics of an object. Abstraction is one of the fundamental elements of the object model.

### **16) Differentiate between a Class and an Object?**

The Object class is the highest-level class in the Java class hierarchy. The Class class is used to represent the classes and interfaces that are loaded by a Java program. The Class class is used to obtain information about an object's design. A Class is only a definition or prototype of real life object. Whereas an object is an instance or living representation of real life object. Every object belongs to a class and every class contains one or more related objects.

### **17) What is meant by Binding?**

Binding denotes association of a name with a class.

### **18) What is meant by Polymorphism?**

Polymorphism literally means taking more than one form. Polymorphism is a characteristic of being able to assign a different behavior or value in a subclass, to something that was declared in a parent class.

### **19) What is Dynamic Binding?**

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding (also known as late binding) means that the code associated with a given procedure call is not known until the time of the call at run-time. It is associated with polymorphism and inheritance.

## 20) What is final modifier?

The final modifier keyword makes that the programmer cannot change the value anymore. The actual meaning depends on whether it is applied to a class, a variable, or a method.

*final* Classes- A final class cannot have subclasses.

*final* Variables- A final variable cannot be changed once it is initialized.

*final* Methods- A final method cannot be overridden by subclasses.

## 21) What is an Interface?

Interface is an outside view of a class or object which emphasizes its abstraction while hiding its structure and secrets of its behaviour.

## 22) What is a base class?

Base class is the most generalised class in a class structure. Most applications have such root classes. In Java, Object is the base class for all classes.

## 23) What is reflection in java?

Reflection allows Java code to discover information about the fields, methods and constructors of loaded classes and to dynamically invoke them.

## 24) Define super class and subclass?

Superclass is a class from which another class inherits. Subclass is a class that inherits from one or more classes.

## 25) What is meant by Binding, Static binding, Dynamic binding?

*Binding*: Binding denotes association of a name with a class

*Static binding*: Static binding is a binding in which the class association is made during compile time. This is also called as *Early* binding.

*Dynamic binding*: Dynamic binding is a binding in which the class association is not made until the object is created at execution time. It is also called as *Late* binding.

## 26) What is reflection API? How are they implemented?

Reflection is the process of introspecting the features and state of a class at runtime and dynamically manipulate at run time. This is supported using Reflection API with built-in classes like Class, Method, Fields, Constructors etc. Example: Using Java Reflection API we can get the class name, by using the getName method.

## 27) What is the difference between a static and a non-static inner class?

A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

**28) What is the difference between abstract class and interface?**

- a) All the methods declared inside an interface are abstract whereas abstract class must have at least one abstract method and others may be concrete or abstract.
- b) In abstract class, key word abstract must be used for the methods whereas interface we need not use that keyword for the methods.
- c) Abstract class must have subclasses whereas interface can't have subclasses.

**29) Can you have an inner class inside a method and what variables can you access?**

Yes, we can have an inner class inside a method and final variables can be accessed.

**30) What is interface and its use?**

Interface is similar to a class which may contain method's signature only but not bodies and it is a formal set of method and constant declarations that must be defined by the class that implements it. Interfaces are useful for:

**31) How is polymorphism achieved in java?**

Inheritance, Overloading and Overriding are used to achieve Polymorphism in java.

**32) What modifiers may be used with top-level class?**

public, abstract and final can be used for top-level class.

**33) What is a cloneable interface and how many methods does it contain?**

It is not having any method because it is a TAGGED or MARKER interface.

**34) What are the methods provided by the object class?**

The Object class provides five methods that are critical when writing multithreaded Java programs:

- notify
- notifyAll
- wait (three versions)