

## **2 Mark Questions and Answer**

### **UNIT-I**

- 1. How could Java classes direct program messages to the system console, but error messages, say to a file?**

The class System has a variable out that represents the standard output, and the variable err that represents the standard error device. By default, they both point at the system console. This how the standard output could be re-directed:

```
Stream st = new Stream(new FileOutputStream("output.txt")); System.setErr(st);  
System.setOut(st);
```

- 2. What's the difference between an interface and an abstract class?**

An abstract class may contain code in method bodies, which is not allowed in an interface. With abstract classes, you have to inherit your class from it and Java does not allow multiple inheritance. On the other hand, you can implement multiple interfaces in your class.

- 3. Why would you use a synchronized block vs. synchronized method?**

Synchronized blocks place locks for shorter periods than synchronized methods.

- 4. Explain the usage of the keyword transient?**

This keyword indicates that the value of this member variable does not have to be serialized with the object. When the class will be de-serialized, this variable will be initialized with a default value of its data type (i.e. zero for integers).

- 5. How can you force garbage collection?**

You can't force GC, but could request it by calling System.gc(). JVM does not guarantee that GC will be started immediately.

- 6. How do you know if an explicit object casting is needed?**

If you assign a superclass object to a variable of a subclass's data type, you need to do explicit casting. For example:

```
Object a; Customer b; b = (Customer) a;
```

When you assign a subclass to a variable having a superclass type, the casting is performed automatically.

- 7. What's the difference between the methods sleep() and wait()**

The code sleep(1000); puts thread aside for exactly one second. The code wait(1000), causes a wait of up to one second. A thread could stop waiting earlier if it receives the notify() or notifyAll() call. The method wait() is defined in the class Object and the method sleep() is defined in the class Thread.

- 8. Can you write a Java class that could be used both as an applet as well as an application?**

Yes. Add a main() method to the applet.

- 9. What's the difference between constructors and other methods?**

Constructors must have the same name as the class and can not return a value. They are only called once while regular methods could be called many times.

**10. Can you call one constructor from another if a class has multiple constructors**

Yes. Use this() syntax.

**11. Explain the usage of Java packages.**

This is a way to organize files when a project consists of multiple modules. It also helps resolve naming conflicts when different packages have classes with the same names. Packages access level also allows you to protect data from being used by the non-authorized classes.

**12. If a class is located in a package, what do you need to change in the OS environment to be able to use it?**

You need to add a directory or a jar file that contains the package directories to the CLASSPATH environment variable. Let's say a class Employee belongs to a package com.xyz.hr; and is located in the file c:\dev\com\xyz\hr\Employee.java. In this case, you'd need to add c:\dev to the variable CLASSPATH. If this class contains the method main(), you could test it from a command prompt window as follows:  
c:\>java com.xyz.hr.Employee

**13. What's the difference between J2SDK 1.5 and J2SDK 5.0?**

There's no difference, Sun Microsystems just re-branded this version.

**14. What would you use to compare two String variables - the operator == or the method equals()?**

A. I'd use the method equals() to compare the values of the Strings and the == to check if two variables point at the same instance of a String object.

**15. Does it matter in what order catch statements for FileNotFoundException and IOException are written?**

Yes, it does. The FileNotFoundException is inherited from the IOException. Exception's subclasses have to be caught first.

**16. Can an inner class declared inside of a method access local variables of this method?**

A. It's possible if these variables are final.

**17. What can go wrong if you replace && with & in the following code:**

**String a=null; if (a!=null && a.length()>10) {...}**

A. A single ampersand here would lead to a NullPointerException.

**18. What's the main difference between a Vector and an ArrayList**

A. Java Vector class is internally synchronized and ArrayList is not.

**19. When should the method invokeLater() be used?**

A. This method is used to ensure that Swing components are updated through the event-dispatching thread.

**20. How can a subclass call a method or a constructor defined in a superclass?**

Use the following syntax: super.myMethod(); To call a constructor of the superclass, just write super(); in the first line of the subclass's constructor  
For senior-level developers:

## UNIT-II

**1. What's the difference between a queue and a stack?**

Stacks works by last-in-first-out rule (LIFO), while queues use the FIFO rule

**2. You can create an abstract class that contains only abstract methods. On the other hand, you can create an interface that declares the same methods. So can you use abstract classes instead of interfaces?**

Sometimes. But your class may be a descendent of another class and in this case the interface is your only option.

**3. What comes to mind when you hear about a young generation in Java?**

Garbage collection.

**4. What comes to mind when someone mentions a shallow copy in Java?**

Object cloning.

**5. If you're overriding the method equals() of an object, which other method you might also consider?**

hashCode()

**6. You are planning to do an indexed search in a list of objects. Which of the two Java collections should you use: ArrayList or LinkedList?**

ArrayList

**7. How would you make a copy of an entire Java object with its state?**

Have this class implement Cloneable interface and call its method clone().

**8. How can you minimize the need of garbage collection and make the memory use more effective?**

Use object pooling and weak object references.

**9. There are two classes: A and B. The class B need to inform a class A when some important event has happened. What Java technique would you use to implement it?**

If these classes are threads I'd consider notify() or notifyAll(). For regular classes you can use the Observer interface.

**10. What access level do you need to specify in the class declaration to ensure that only classes from the same directory can access it?**

You do not need to specify any access level, and Java will use a default package access level.

**11. What is garbage collection? What is the process that is responsible for doing that in java? –**

Reclaiming the unused memory by the invalid objects. Garbage collector is responsible for this process

**12. What kind of thread is the Garbage collector thread? - It is a daemon thread.**

**13.What is a daemon thread? - These are the threads which can run without user intervention. The JVM can exit when there are daemon thread by killing them abruptly.**

**14. How will you invoke any external process in Java? -**

`Runtime.getRuntime().exec(...)`

**15. What is the finalize method do?** - Before the invalid objects get garbage collected, the JVM give the user a chance to clean up some resources before it got garbage collected.

**16. What is mutable object and immutable object?** - If a object value is changeable then we can call it as Mutable object. (Ex., StringBuffer, ...) If you are not allowed to change the value of an object, it is immutable object. (Ex., String, Integer, Float, ...)

**17. What is the basic difference between string and stringbuffer object?** - String is an immutable object. StringBuffer is a mutable object.

**18. What is the purpose of Void class?** - The Void class is an uninstantiable placeholder class to hold a reference to the Class object representing the primitive Java type void.

**19. What is reflection?** - Reflection allows programmatic access to information about the fields, methods and constructors of loaded classes, and the use reflected fields, methods, and constructors to operate on their underlying counterparts on objects, within security restrictions.

**20. What is the base class for Error and Exception?** - Throwable

## UNIT-III

**1.What is the byte range? & What is the implementation of destroy method in java.. is it native or java code? -**

128 to 127 .This method is not implemented.

**2.What is a package? –**

To group set of classes into a single unit is known as packaging. Packages provides wide namespace ability.

**3.What are the approaches that you will follow for making a program very efficient?** - By avoiding too much of static methods avoiding the excessive and unnecessary use of synchronized methods Selection of related classes based on the application (meaning synchronized classes for multiuser and non-synchronized classes for single user) Usage of appropriate design patterns Using cache methodologies for remote invocations Avoiding creation of variables within a loop and lot more.

**4. What is Locale?** - A Locale object represents a specific geographical, political, or cultural region

**5. How will you load a specific locale?** - Using `ResourceBundle.getBundle(...)`;

**6. What is JIT and its use?** - Really, just a very fast compiler... In this incarnation, pretty much a one-pass compiler — no offline computations. So you can't look at the whole method, rank the expressions according to which ones are re-used the most, and then generate code. In theory terms, it's an on-line problem.

**7. Is JVM a compiler or an interpreter?** - Interpreter

**8. When you think about optimization, what is the best way to find out the time/memory consuming process?** - Using profiler

**9. What is the purpose of assert keyword used in JDK1.4.x?** - In order to validate certain expressions. It effectively replaces the if block and automatically throws the AssertionError on failure. This keyword should be used for the critical arguments. Meaning, without that the method does nothing.

**10. How will you get the platform dependent values like line separator, path separator, etc., ?**

Using `System.getProperty(...)` (line.separator, path.separator, ...)

**11. What is skeleton and stub? what is the purpose of those?**

Stub is a client side representation of the server, which takes care of communicating with the remote server. Skeleton is the server side representation. But that is no more in use... it is deprecated long before in JDK.

**12. What is the final keyword denotes?** - final keyword denotes that it is the final implementation for that method or variable or class. You can't override that method/variable/class any more.

**13. What is the significance of ListIterator?** - You can iterate back and forth.

**14. What is the major difference between LinkedList and ArrayList?** - LinkedList are meant for sequential accessing. ArrayList are meant for random accessing.

**15. What is nested class?**

If all the methods of an inner class are static then it is a nested class.

**16. What is inner class?**

If the methods of the inner class can only be accessed via the instance of the inner class, then it is called inner class.

**17. What is composition?**

Holding the reference of the other class within some other class is known as composition.

**18. What is aggregation?**

It is a special type of composition. If you expose all the methods of a composite class and route the method call to the composite method through its reference, then it is called aggregation.

**19. What are the methods in Object?**

clone, equals, wait, finalize, getClass, hashCode, notify, notifyAll, toString

**21. Can you instantiate the Math class?**

You can't instantiate the math class. All the methods in this class are static. And the constructor is not public.

## 22. What is singleton?

It is one of the design patterns. This falls in the creational pattern of the design pattern. There will be only one instance for that entire JVM. You can achieve this by having the private constructor in the class. For eg., public class Singleton { private static final Singleton s = new Singleton(); private Singleton() { } public static Singleton getInstance() { return s; } // all non static methods ... }

## 23. What is DriverManager?

The basic service to manage set of JDBC drivers.

## 24. What is Class.forName() does and how it is useful?

It loads the class into the ClassLoader. It returns the Class. Using that you can get the instance ( "class-instance".newInstance() ).

## 25. What is a DatabaseMetaData? –

26. Comprehensive information about the database as a whole.

# UNIT-IV

## 1. How could Java classes direct program messages to the system console, but error messages, say to a file?

The class System has a variable out that represents the standard output, and the variable err that represents the standard error device. By default, they both point at the system console. This how the standard output could be re-directed:

```
Stream st =  
new Stream (new  
    FileOutputStream ("techinterviews_com.txt"));  
System.setErr(st);  
System.setOut(st);
```

## 2. What's the difference between an interface and an abstract class?

An abstract class may contain code in method bodies, which is not allowed in an interface. With abstract classes, you have to inherit your class from it and Java does not allow multiple inheritance. On the other hand, you can implement multiple interfaces in your class.

## 3. Why would you use a synchronized block vs. synchronized method?

Synchronized blocks place locks for shorter periods than synchronized methods.

## 4. Explain the usage of the keyword transient?

This keyword indicates that the value of this member variable does not have to be serialized with the object. When the class will be de-serialized, this variable will be initialized with a default value of its data type (i.e. zero for integers).

## 5. How can you force garbage collection?

You can't force GC, but could request it by calling System.gc(). JVM does not guarantee that GC will be started immediately.

## 6. How do you know if an explicit object casting is needed?

If you assign a superclass object to a variable of a subclass's data type, you need to do explicit casting. For example:

```
Object a; Customer b; b = (Customer) a;
```

When you assign a subclass to a variable having a superclass type, the casting is performed automatically.

**7. What's the difference between the methods sleep() and wait()**

The code sleep(1000); puts thread aside for exactly one second. The code wait(1000), causes a wait of up to one second. A thread could stop waiting earlier if it receives the notify() or notifyAll() call. The method wait() is defined in the class Object and the method sleep() is defined in the class Thread.

**8. Can you write a Java class that could be used both as an applet as well as an application?**

Yes. Add a main() method to the applet.

**9. What's the difference between constructors and other methods?**

Constructors must have the same name as the class and can not return a value. They are only called once while regular methods could be called many times.

**10. Can you call one constructor from another if a class has multiple constructors**

Yes. Use this() syntax.

**11. Explain the usage of Java packages.**

This is a way to organize files when a project consists of multiple modules. It also helps resolve naming conflicts when different packages have classes with the same names. Packages access level also allows you to protect data from being used by the non-authorized classes.

**12. If a class is located in a package, what do you need to change in the OS environment to be able to use it?**

You need to add a directory or a jar file that contains the package directories to the CLASSPATH environment variable. Let's say a class Employee belongs to a package com.xyz.hr; and is located in the file c:/dev/com.xyz.hr.Employee.java. In this case, you'd need to add c:/dev to the variable CLASSPATH. If this class contains the method main(), you could test it from a command prompt window as follows:

```
c:\>java com.xyz.hr.Employee
```

**13. What's the difference between J2SDK 1.5 and J2SDK 5.0?**

There's no difference, Sun Microsystems just re-branded this version.

**14. What would you use to compare two String variables - the operator == or the method equals()?**

I'd use the method equals() to compare the values of the Strings and the == to check if two variables point at the same instance of a String object.

**15. Does it matter in what order catch statements for FileNotFoundException and IOException are written?**

A. Yes, it does. The FileNotFoundException is inherited from the IOException. Exception's subclasses have to be caught first.

**16. Can an inner class declared inside of a method access local variables of this method?**

It's possible if these variables are final.

**17. What can go wrong if you replace && with & in the following code:**

```
String a=null;
```

```
if (a!=null && a.length()>10)
```

```
{...}
```

A single ampersand here would lead to a NullPointerException.

## UNIT-V

1. **Do I need to use synchronized on setValue(int)?** - It depends whether the method affects method local variables, class static or instance variables. If only method local variables are changed, the value is said to be confined by the method and is not prone to threading issues.
2. **Do I need to use synchronized on setValue(int)?** - It depends whether the method affects method local variables, class static or instance variables. If only method local variables are changed, the value is said to be confined by the method and is not prone to threading issues.
3. **What is the SwingUtilities.invokeLater(Runnable) method for?** - The static utility method invokeLater(Runnable) is intended to execute a new runnable thread from a Swing application without disturbing the normal sequence of event dispatching from the Graphical User Interface (GUI). The method places the runnable object in the queue of Abstract Windowing Toolkit (AWT) events that are due to be processed and returns immediately. The runnable object's run() method is only called when it reaches the front of the queue. The deferred effect of the invokeLater(Runnable) method ensures that any necessary updates to the user interface can occur immediately, and the runnable work will begin as soon as those high priority events are dealt with. The invoke later method might be used to start work in response to a button click that also requires a significant change to the user interface, perhaps to restrict other activities, while the runnable thread executes.
4. **What is the volatile modifier for?** - The volatile modifier is used to identify variables whose values should not be optimized by the Java Virtual Machine, by caching the value for example. The volatile modifier is typically used for variables that may be accessed or modified by numerous independent threads and signifies that the value may change without synchronization.
5. **Which class is the wait() method defined in?** - The wait() method is defined in the Object class, which is the ultimate superclass of all others. So the Thread class and any Runnable implementation inherit this method from Object. The wait() method is normally called on an object in a multi-threaded program to allow other threads to run. The method should only be called by a thread that has ownership of the object's monitor, which usually means it is in a synchronized method or statement block.
6. **What is a working thread?** - A working thread, more commonly known as a worker thread is the key part of a design pattern that allocates one thread to execute one task. When the task is complete, the thread may return to a thread pool for later use. In this scheme a thread may execute arbitrary tasks, which are passed in the form of a Runnable method argument, typically execute(Runnable). The runnable tasks are usually stored in a queue until a thread host is available to run them. The worker thread design pattern is usually used to handle many concurrent tasks where it is not important which finishes first and no single task needs to be coordinated with another. The task queue controls how many threads run concurrently to improve the overall



Adhiparasakthi College of Engineering, G.B.Nagar, Kalavai  
performance of the system. However, a worker thread framework requires relatively complex programming to set up, so should not be used where simpler threading techniques can achieve similar results.

- 7. What is a green thread?** - A green thread refers to a mode of operation for the Java Virtual Machine (JVM) in which all code is executed in a single operating system thread. If the Java program has any concurrent threads, the JVM manages multi-threading internally rather than using other operating system threads. There is a significant processing overhead for the JVM to keep track of thread states and swap between them, so green thread mode has been deprecated and removed from more recent Java implementations. Current JVM implementations make more efficient use of native operating system threads.
- 8. What are native operating system threads?** - Native operating system threads are those provided by the computer operating system that plays host to a Java application, be it Windows, Mac or GNU/Linux. Operating system threads enable computers to run many programs simultaneously on the same central processing unit (CPU) without clashing over the use of system resources or spending lots of time running one program at the expense of another. Operating system thread management is usually optimised to specific microprocessor architecture and features so that it operates much faster than Java green thread
- 9. What's the main difference between a Vector and an ArrayList**  
Java Vector class is internally synchronized and ArrayList is not.
- 10. When should the method invokeLater() be used?**  
This method is used to ensure that Swing components are updated through the event-dispatching thread.
- 11. How can a subclass call a method or a constructor defined in a superclass?**  
Use the following syntax: super.myMethod(); To call a constructor of the superclass, just write super(); in the first line of the subclass's constructor.
- 12. What's the difference between a queue and a stack?**  
Stacks works by last-in-first-out rule (LIFO), while queues use the FIFO rule.
- 13. You can create an abstract class that contains only abstract methods. On the other hand, you can create an interface that declares the same methods. So can you use abstract classes instead of interfaces?**  
Sometimes. But your class may be a descendent of another class and in this case the interface is your only option.
- 14. What comes to mind when you hear about a young generation in Java?**  
Garbage collection.
- 15. What comes to mind when someone mentions a shallow copy in Java?**  
Object cloning.
- 16. If you're overriding the method equals() of an object, which other method you might also consider?**  
hashCode()

**17. You are planning to do an indexed search in a list of objects. Which of the two Java collections should you use: ArrayList or LinkedList?**

ArrayList

**18. How would you make a copy of an entire Java object with its state?**

Have this class implement Cloneable interface and call its method clone().

**19. How can you minimize the need of garbage collection and make the memory use more effective?**

Use object pooling and weak object references.

**20. There are two classes: A and B. The class B need to inform a class A when some important event has happened. What Java technique would you use to implement it?**

If these classes are threads I'd consider notify() or notifyAll(). For regular classes you can use the Observer interface.

**21. What access level do you need to specify in the class declaration to ensure that only classes from the same directory can access it?**

You do not need to specify any access level, and Java will use a default package access level.