**Q: What is the difference between an Interface and an Abstract class?**

**A:** An abstract class can have instance methods that implement a default behavior. An Interface can only declare constants and instance methods, but cannot implement default behavior and all methods are implicitly abstract. An interface has all public members and no implementation. An abstract class is a class, which may have the usual flavors of class members (private, protected, etc.), but has some abstract methods.

.

TOP

**Q: What is the purpose of garbage collection in Java, and when is it used?**

**A:** 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.

TOP

**Q: Describe synchronization in respect to multithreading.**

**A:** 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.

TOP

**Q: Explain different way of using thread?**

**A:** The thread could be implemented by using runnable interface or by inheriting from the Thread class. The former is more advantageous, 'cause when you are going for multiple inheritance..the only interface can help.

TOP

**Q: What are pass by reference and passby value?**

**A:** 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.

TOP

**Q: What is HashMap and Map?**

**A:** Map is Interface and Hashmap is class that implements that.

TOP

**What are Encapsulation, Inheritance and Polymorphism?**- Encapsulation is the mechanism that binds together code and data it manipulates and keeps both safe from outside interference and misuse. Inheritance is the process by which one object acquires the properties of another object. Polymorphism is the feature that allows one interface to be used for general class actions

**Q: Difference between HashMap and HashTable?**

**A:** 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.

TOP

**Q: Difference between Vector and ArrayList?**

**A:** Vector is synchronized whereas arraylist is not.

TOP

**Q: Difference between Swing and Awt?**

**A:** AWT are heavy-weight componenets. Swings are light-weight components. Hence swing works faster than AWT.

TOP

**Q: What is the difference between a constructor and a method?**

**A:** 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.

TOP

**Q: What is an Iterator?**

**A:** 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.

TOP

**Q: State the significance of public, private, protected, default modifiers both singly and in combination and state the effect of package relationships on declared items qualified by these modifiers.**

**A:** ***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.

TOP

**Q: What is an abstract class?**

**A:** 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.

TOP

**Q: What is static in java?**

**A:** 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.

TOP

**Q: What is final?**

**A:** 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).

**Q: What if the main method is declared as private?**

**A:** The program compiles properly but at runtime it will give "Main method not public." message.
[ Received from Sandesh Sadhale]

TOP

**Q: What if the static modifier is removed from the signature of the main method?**

**A:** Program compiles. But at runtime throws an error "NoSuchMethodError".
[ Received from Sandesh Sadhale]

TOP

**Q: What if I write static public void instead of public static void?**

**A:** Program compiles and runs properly.
[ Received from Sandesh Sadhale]

TOP

**Q: What if I do not provide the String array as the argument to the method?**

**A:** Program compiles but throws a runtime error "NoSuchMethodError".
[ Received from Sandesh Sadhale]

TOP

**Q: What is the first argument of the String array in main method?**

**A:** The String array is empty. It does not have any element. This is unlike C/C++ where the first element by default is the program name.
[ Received from Sandesh Sadhale]

TOP

**Q: If I do not provide any arguments on the command line, then the String array of Main method will be empty or null?**

**A:** It is empty. But not null.
[ Received from Sandesh Sadhale] TOP

**Q: How can one prove that the array is not null but empty using one line of code?**
**A:** Print args.length. It will print 0. That means it is empty. But if it would have been null then it would have thrown a NullPointerException on attempting to print args.length.
[ Received from Sandesh Sadhale] TOP

**Q: What environment variables do I need to set on my machine in order to be able to run Java programs?**
**A:** CLASSPATH and PATH are the two variables.
[ Received from Sandesh Sadhale] TOP

**Q: Can an application have multiple classes having main method?**
**A:** Yes it is possible. While starting the application we mention the class name to be run. The JVM will look for the Main method only in the class whose name you have mentioned. Hence there is not conflict amongst the multiple classes having main method.
[ Received from Sandesh Sadhale] TOP

**Q: Can I have multiple main methods in the same class?**
**A:** No the program fails to compile. The compiler says that the main method is already defined in the class.
[ Received from Sandesh Sadhale] TOP

**Q: Do I need to import java.lang package any time? Why ?**
**A:** No. It is by default loaded internally by the JVM.
[ Received from Sandesh Sadhale] TOP

**Q: Can I import same package/class twice? Will the JVM load the package twice at runtime?**
**A:** One can import the same package or same class multiple times. Neither compiler nor JVM complains abt it. And the JVM will internally load the class only once no matter how many times you import the same class.
[ Received from Sandesh Sadhale] TOP

**Q: What are Checked and UnChecked Exception?**
**A:** A checked exception is some subclass of Exception (or Exception itself), excluding class RuntimeException and its subclasses.
Making an exception checked forces client programmers to deal with the possibility that the exception will be thrown. eg, IOException thrown by java.io.FileInputStream's read() method·
Unchecked exceptions are RuntimeException and any of its subclasses. Class Error and its subclasses also are unchecked. With an unchecked exception, however, the compiler doesn't force client programmers either to catch the exception or declare it in a throws clause. In fact, client programmers may not even know that the exception could be thrown. eg, StringIndexOutOfBoundsException thrown by String's charAt() method· Checked exceptions must be caught at compile time. Runtime exceptions do not need to

be. Errors often cannot be.

TOP

**Q: What is Overriding?**

**A:** When a class defines a method using the same name, return type, and arguments as a method in its superclass, the method in the class overrides the method in the superclass.
When the method is invoked for an object of the class, it is the new definition of the method that is called, and not the method definition from superclass. Methods may be overridden to be more public, not more private.

TOP

**Q: What are different types of inner classes?**

**A:** *Nested top-level classes*, *Member classes, Local classes, Anonymous classes*

***Nested top-level classes***- If you declare a class within a class and specify the static modifier, the compiler treats the class just like any other top-level class. Any class outside the declaring class accesses the nested class with the declaring class name acting similarly to a package. eg, outer.inner. Top-level inner classes implicitly have access only to static variables.There can also be inner interfaces. All of these are of the nested top-level variety.

***Member classes*** - Member inner classes are just like other member methods and member variables and access to the member class is restricted, just like methods and variables. This means a public member class acts similarly to a nested top-level class. The primary difference between member classes and nested top-level classes is that member classes have access to the specific instance of the enclosing class.

***Local classes*** - Local classes are like local variables, specific to a block of code. Their visibility is only within the block of their declaration. In order for the class to be useful beyond the declaration block, it would need to implement a more publicly available interface.Because local classes are not members, the modifiers public, protected, private, and static are not usable.

***Anonymous classes*** - Anonymous inner classes extend local inner classes one level further. As anonymous classes have no name, you cannot provide a constructor.

**Q: Are the imports checked for validity at compile time? e.g. will the code containing an impo**
**java.lang.ABCD compile?**

**A:** Yes the imports are checked for the semantic validity at compile time. The code containing above lin
compile. It will throw an error saying,can not resolve symbol
symbol : class ABCD
location: package io
import java.io.ABCD;
[ Received from Sandesh Sadhale]

**Q: Does importing a package imports the subpackages as well? e.g. Does importing com.MyTest.* also import com.MyTest.UnitTests.*?**

**A:** No you will have to import the subpackages explicitly. Importing com.MyTest.* will import classes in the package MyTest only. It will not import any class in any of it's subpackage.

[ Received from Sandesh Sadhale] TOP

**Q: What is the difference between declaring a variable and defining a variable?**

**A:** In declaration we just mention the type of the variable and it's name. We do not initialize it. But defining means declaration + initialization.
e.g String s; is just a declaration while String s = new String ("abcd"); Or String s = "abcd"; are both definitions.

[ Received from Sandesh Sadhale] TOP

**Q: What is the default value of an object reference declared as an instance variable?**

**A:** null unless we define it explicitly.

[ Received from Sandesh Sadhale] TOP

**Q: Can a top level class be private or protected?**

**A:** No. A top level class can not be private or protected. It can have either "public" or no modifier. If it does not have a modifier it is supposed to have a default access.If a top level class is declared as private the compiler will complain that the "modifier private is not allowed here". This means that a top level class can not be private. Same is the case with protected.

[ Received from Sandesh Sadhale] TOP

**Q: What type of parameter passing does Java support?**

**A:** In Java the arguments are always passed by value .

[ Update from Eki and Jyothish Venu] TOP

**Q: Primitive data types are passed by reference or pass by value?**

**A:** Primitive data types are passed by value.

[ Received from Sandesh Sadhale] TOP

**Q: Objects are passed by value or by reference?**

**A:** Java only supports pass by value. With objects, the object reference itself is passed by value and so both the original reference and parameter copy both refer to the same object .

[ Update from Eki and Jyothish Venu] TOP

**Q: What is serialization?**

**A:** Serialization is a mechanism by which you can save the state of an object by converting it to a byte stream.

[ Received from Sandesh Sadhale] TOP

**Q: How do I serialize an object to a file?**

**A:** The class whose instances are to be serialized should implement an interface Serializable. Then you pass the instance to the ObjectOutputStream which is connected to a fileoutputstream. This will save the object to a file.

[ Received from Sandesh Sadhale] TOP

**Q: Which methods of Serializable interface should I implement?**
**A:** The serializable interface is an empty interface, it does not contain any methods. So we do not implement any methods.
[ Received from Sandesh Sadhale] TOP

**Q: How can I customize the seralization process? i.e. how can one have a control over the serialization process?**
**A:** Yes it is possible to have control over serialization process. The class should implement Externalizable interface. This interface contains two methods namely readExternal and writeExternal. You should implement these methods and write the logic for customizing the serialization process.
[ Received from Sandesh Sadhale] TOP

**Q: What is the common usage of serialization?**
**A:** Whenever an object is to be sent over the network, objects need to be serialized. Moreover if the state of an object is to be saved, objects need to be serilazed.
[ Received from Sandesh Sadhale] TOP

**Q: What is Externalizable interface?**
**A:** Externalizable is an interface which contains two methods readExternal and writeExternal. These methods give you a control over the serialization mechanism. Thus if your class implements this interface, you can customize the serialization process by implementing these methods.
[ Received from Sandesh Sadhale] TOP

**Q: When you serialize an object, what happens to the object references included in the object?**
**A:** The serialization mechanism generates an object graph for serialization. Thus it determines whether the included object references are serializable or not. This is a recursive process. Thus when an object is serialized, all the included objects are also serialized alongwith the original obect.
[ Received from Sandesh Sadhale] TOP

**Q: What one should take care of while serializing the object?**
**A:** One should make sure that all the included objects are also serializable. If any of the objects is not serializable then it throws a NotSerializableException.
[ Received from Sandesh Sadhale] TOP

**Q: What happens to the static fields of a class during serialization?**
**A:** There are three exceptions in which serialization doesnot necessarily read and write to the stream. These are
1. Serialization ignores static fields, because they are not part of ay particular state state.
2. Base class fields are only hendled if the base class itself is serializable.
3. Transient fields.
[ Received from Sandesh Sadhale Modified after P.John David comments.] TOP

**Q: Does Java provide any construct to find out the size of an object?**
**A:** No there is not sizeof operator in Java. So there is not direct way to determine the

size of an object directly in Java.
[ Received from Sandesh Sadhale] TOP

**Q: Give a simplest way to find out the time a method takes for execution without using any profiling tool?**

**A:** Read the system time just before the method is invoked and immediately after method returns. Take the time difference, which will give you the time taken by a method for execution.

To put it in code...

```
long start = System.currentTimeMillis ();
method ();
long end = System.currentTimeMillis ();

System.out.println ("Time taken for execution is " + (end - start));
```

Remember that if the time taken for execution is too small, it might show that it is taking zero milliseconds for execution. Try it on a method which is big enough, in the sense the one which is doing considerable amout of processing.
[ Received from Sandesh Sadhale] TOP

**Q: What are wrapper classes?**

**A:** Java provides specialized classes corresponding to each of the primitive data types. These are called wrapper classes. They are e.g. Integer, Character, Double etc.
[ Received from Sandesh Sadhale] TOP

**Q: Why do we need wrapper classes?**

**A:** It is sometimes easier to deal with primitives as objects. Moreover most of the collection classes store objects and not primitive data types. And also the wrapper classes provide many utility methods also. Because of these resons we need wrapper classes. And since we create instances of these classes we can store them in any of the collection classes and pass them around as a collection. Also we can pass them around as method parameters where a method expects an object.
[ Received from Sandesh Sadhale] TOP

**Q: What are checked exceptions?**

**A:** Checked exception are those which the Java compiler forces you to catch. e.g. IOException are checked Exceptions.
[ Received from Sandesh Sadhale] TOP

**Q: What are runtime exceptions?**

**A:** Runtime exceptions are those exceptions that are thrown at runtime because of either wrong input data or because of wrong business logic etc. These are not checked by the compiler at compile time.
[ Received from Sandesh Sadhale] TOP

**Q: What is the difference between error and an exception?**

**A:** An error is an irrecoverable condition occurring at runtime. Such as OutOfMemory

error. These JVM errors and you can not repair them at runtime. While exceptions are conditions that occur because of bad input etc. e.g. FileNotFoundException will be thrown if the specified file does not exist. Or a NullPointerException will take place if you try using a null reference. In most of the cases it is possible to recover from an exception (probably by giving user a feedback for entering proper values etc.).

[ Received from Sandesh Sadhale] TOP

**Q: How to create custom exceptions?**

**A:** Your class should extend class Exception, or some more specific type thereof.

[ Received from Sandesh Sadhale] TOP

**Q: If I want an object of my class to be thrown as an exception object, what should I do?**

**A:** The class should extend from Exception class. Or you can extend your class from some more precise exception type also.

[ Received from Sandesh Sadhale] TOP

**Q: If my class already extends from some other class what should I do if I want an instance of my class to be thrown as an exception object?**

**A:** One can not do anytihng in this scenarion. Because Java does not allow multiple inheritance and does not provide any exception interface as well.

[ Received from Sandesh Sadhale] TOP

**Q: How does an exception permeate through the code?**

**A:** An unhandled exception moves up the method stack in search of a matching When an exception is thrown from a code which is wrapped in a try block followed by one or more catch blocks, a search is made for matching catch block. If a matching type is found then that block will be invoked. If a matching type is not found then the exception moves up the method stack and reaches the caller method. Same procedure is repeated if the caller method is included in a try catch block. This process continues until a catch block handling the appropriate type of exception is found. If it does not find such a block then finally the program terminates.

[ Received from Sandesh Sadhale] TOP

**Q: What are the different ways to handle exceptions?**

**A:** There are two ways to handle exceptions,

1. By wrapping the desired code in a try block followed by a catch block to catch the exceptions. and

2. List the desired exceptions in the throws clause of the method and let the caller of the method hadle those exceptions.

[ Received from Sandesh Sadhale] TOP

**Q: What is the basic difference between the 2 approaches to exception handling.**
**1> try catch block and**
**2> specifying the candidate exceptions in the throws clause?**
**When should you use which approach?**

**A:** In the first approach as a programmer of the method, you urself are dealing with the exception. This is fine if you are in a best position to decide should be done in

case of an exception. Whereas if it is not the responsibility of the method to deal with it's own exceptions, then do not use this approach. In this case use the second approach. In the second approach we are forcing the caller of the method to catch the exceptions, that the method is likely to throw. This is often the approach library creators use. They list the exception in the throws clause and we must catch them. You will find the same approach throughout the java libraries we use.
[ Received from Sandesh Sadhale] TOP

**Q: Is it necessary that each try block must be followed by a catch block?**
**A:** It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.
[ Received from Sandesh Sadhale] TOP

**Q: If I write return at the end of the try block, will the finally block still execute?**
**A:** Yes even if you write return as the last statement in the try block and no exception occurs, the finally block will execute. The finally block will execute and then the control return.
[ Received from Sandesh Sadhale] TOP

**Q: If I write System.exit (0); at the end of the try block, will the finally block still execute?**
**A:** No in this case the finally block will not execute because when you say System.exit (0); the control immediately goes out of the program, and thus finally never executes.
[ Received from Sandesh Sadhale] TOP

**Q: How are Observer and Observable used?**
**A:** Objects that subclass the Observable class maintain a list of observers. When an Observable object is updated it invokes the update() method of each of its observers to notify the observers that it has changed state. The Observer interface is implemented by objects that observe Observable objects.
[Received from Venkateswara Manam] TOP

**Q: What is synchronization and why is it important?**
**A:** With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often leads to significant errors.
[ Received from Venkateswara Manam] TOP

**Q: How does Java handle integer overflows and underflows?**
**A:** It uses those low order bytes of the result that can fit into the size of the type allowed by the operation.

[ Received from Venkateswara Manam] TOP

**Q: Does garbage collection guarantee that a program will not run out of memory?**

**A:** Garbage collection does not guarantee that a program will not run out of memory. It is possible for programs to use up memory resources faster than they are garbage collected. It is also possible for programs to create objects that are not subject to garbage collection

.

[ Received from Venkateswara Manam] TOP

**Q: What is the difference between preemptive scheduling and time slicing?**

**A:** Under preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence. Under time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks. The scheduler then determines which task should execute next, based on priority and other factors.

[ Received from Venkateswara Manam] TOP

**Q: When a thread is created and started, what is its initial state?**

**A:** A thread is in the ready state after it has been created and started.

[ Received from Venkateswara Manam] TOP

**Q: What is the purpose of finalization?**

**A:** The purpose of finalization is to give an unreachable object the opportunity to perform any cleanup processing before the object is garbage collected.

[ Received from Venkateswara Manam] TOP

**Q: What is the Locale class?**

**A:** The Locale class is used to tailor program output to the conventions of a particular geographic, political, or cultural region.

[ Received from Venkateswara Manam] TOP

**Q: What is the difference between a while statement and a do statement?**

**A:** A while statement checks at the beginning of a loop to see whether the next loop iteration should occur. A do statement checks at the end of a loop to see whether the next iteration of a loop should occur. The do statement will always execute the body of a loop at least once.

[ Received from Venkateswara Manam] TOP

**Q: What is the difference between static and non-static variables?**

**A:** A static variable is associated with the class as a whole rather than with specific instances of a class. Non-static variables take on unique values with each object instance.

[ Received from Venkateswara Manam] TOP

**Q: How are this() and super() used with constructors?**

**A:** This() is used to invoke a constructor of the same class. super() is used to invoke a superclass constructor.

[ Received from Venkateswara Manam] TOP

**Q: What are synchronized methods and synchronized statements?**

**A:** Synchronized methods are methods that are used to control access to an object. A thread only executes a synchronized method after it has acquired the lock for the method's object or class. Synchronized statements are similar to synchronized methods. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement.
[ Received from Venkateswara Manam] TOP

**Q: What is daemon thread and which method is used to create the daemon thread?**

**A:** Daemon thread is a low priority thread which runs intermittently in the back ground doing the garbage collection operation for the java runtime system. setDaemon method is used to create a daemon thread.
[ Received from Shipra Kamra] TOP

**Q: Can applets communicate with each other?**

**A:** At this point in time applets may communicate with other applets running in the same virtual machine. If the applets are of the same class, they can communicate via shared static variables. If the applets are of different classes, then each will need a reference to the same class with static variables. In any case the basic idea is to pass the information back and forth through a static variable.

An applet can also get references to all other applets on the same page using the getApplets() method of java.applet.AppletContext. Once you get the reference to an applet, you can communicate with it by using its public members.

It is conceivable to have applets in different virtual machines that talk to a server somewhere on the Internet and store any data that needs to be serialized there. Then, when another applet needs this data, it could connect to this same server. Implementing this is non-trivial.
[ Received from Krishna Kumar ] TOP

**Q: What are the steps in the JDBC connection?**

**A:** While making a JDBC connection we go through the following steps :

Step 1 : Register the database driver by using :

Class.forName(\" driver classs for that specific database\" );

Step 2 : Now create a database connection using :

Connection con = DriverManager.getConnection(url,username,password);

Step 3: Now Create a query using :

Statement stmt = Connection.Statement(\"select * from TABLE NAME\");

Step 4 : Exceute the query :

stmt.exceuteUpdate();

[ Received from Shri Prakash Kunwar] TOP

**Q: How does a try statement determine which catch clause should be used to handle an exception?**

**A:** When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exceptionis executed. The remaining catch clauses are ignored.

**Q: Can an unreachable object become reachable again?**

**A:** An unreachable object may become reachable again. This can happen when the object's finalize() method is invoked and the object performs an operation which causes it to become accessible to reachable objects.
[Received from P Rajesh] TOP

**Q: What method must be implemented by all threads?**

**A:** All tasks must implement the run() method, whether they are a subclass of Thread or implement the Runnable interface.
[ Received from P Rajesh] TOP

**Q: What are synchronized methods and synchronized statements?**

**A:** Synchronized methods are methods that are used to control access to an object. A thread only executes a synchronized method after it has acquired the lock for the method's object or class. Synchronized statements are similar to synchronized methods. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement.
[ Received from P Rajesh] TOP

**Q: What is Externalizable?**

**A:** Externalizable is an Interface that extends Serializable Interface. And sends data into Streams in Compressed Format. It has two methods, writeExternal(ObjectOuput out) and readExternal(ObjectInput in)
[ Received from Venkateswara Manam] TOP

**Q: What modifiers are allowed for methods in an Interface?**

**A:** Only public and abstract modifiers are allowed for methods in interfaces.
[ Received from P Rajesh] TOP

**Q: What are some alternatives to inheritance?**

**A:** Delegation is an alternative to inheritance. Delegation means that you include an instance of another class as an instance variable, and forward messages to the instance. It is often safer than inheritance because it forces you to think about each message you forward, because the instance is of a known class, rather than a new class, and because it doesn't force you to accept all the methods of the super class: you can provide only the methods that really make sense. On the other hand, it makes you write more code, and it is harder to re-use (because it is not a subclass).
[ Received from P Rajesh] TOP

**Q: What does it mean that a method or field is "static"?**

**A:** Static variables and methods are instantiated only once per class. In other words they are class variables, not instance variables. If you change the value of a static variable in a particular object, the value of that variable changes for all instances of that class.

Static methods can be referenced with the name of the class rather than the name of a particular object of the class (though that works too). That's how library methods like System.out.println() work out is a static field in the java.lang.System class.

[ Received from P Rajesh] TOP

**Q: What is the difference between preemptive scheduling and time slicing?**

**A:** Under preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence. Under time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks. The scheduler then determines which task should execute next, based on priority and other factors.

[ Received from P Rajesh] TOP

**Q: What is the catch or declare rule for method declarations?**

**A:** If a checked exception may be thrown within the body of a method, the method must either catch the exception or declare it in its throws clause.

[ Received from P Rajesh]

**Q: How are Observer and Observable used?**

**A:** Objects that subclass the Observable class maintain a list of observers. When an Observable object is updated it invokes the update() method of each of its observers to notify the observers that it has changed state. The Observer interface is implemented by objects that observe Observable objects.

[Received from Venkateswara Manam] TOP

**Q: What is synchronization and why is it important?**

**A:** With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often leads to significant errors.

[ Received from Venkateswara Manam] TOP

[ Received from Shri Prakash Kunwar] TOP

**Q: How does a try statement determine which catch clause should be used to handle an exception?**

**A:** When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exceptionis executed. The remaining catch clauses are ignored.

**Q: Is Empty .java file a valid source file?**
**A:** Yes, an empty .java file is a perfectly valid source file.
[Received from Sandesh Sadhale] TOP

**Q: Can a .java file contain more than one java classes?**
**A:** Yes, a .java file contain more than one java classes, provided at the most one of them is a public class.
[ Received from Sandesh Sadhale] TOP

**Q: Is String a primitive data type in Java?**
**A:** No String is not a primitive data type in Java, even though it is one of the most extensively used object. Strings in Java are instances of String class defined in java.lang package.
[ Received from Sandesh Sadhale] TOP

**Q: Is main a keyword in Java?**
**A:** No, main is not a keyword in Java.
[ Received from Sandesh Sadhale] TOP

**Q: Is next a keyword in Java?**
**A:** No, next is not a keyword.
[ Received from Sandesh Sadhale] TOP

**Q: Is delete a keyword in Java?**
**A:** No, delete is not a keyword in Java. Java does not make use of explicit destructors the way C++ does.
[ Received from Sandesh Sadhale] TOP

**Q: Is exit a keyword in Java?**
**A:** No. To exit a program explicitly you use exit method in System object.
[ Received from Sandesh Sadhale] TOP

**Q: What happens if you dont initialize an instance variable of any of the primitive types in Java?**
**A:** Java by default initializes it to the default value for that primitive type. Thus an int will be initialized to 0, a boolean will be initialized to false.
[ Received from Sandesh Sadhale] TOP

**Q: What will be the initial value of an object reference which is defined as an instance variable?**
**A:** The object references are all initialized to null in Java. However in order to do anything useful with these references, you must set them to a valid object, else you will get NullPointerExceptions everywhere you try to use such default initialized references.
[ Received from Sandesh Sadhale] TOP

**Q: What are the different scopes for Java variables?**
**A:** The scope of a Java variable is determined by the context in which the variable is declared. Thus a java variable can have one of the three scopes at any given point

in time.
1. Instance : - These are typical object level variables, they are initialized to default values at the time of creation of object, and remain accessible as long as the object accessible.
2. Local : - These are the variables that are defined within a method. They remain accessbile only during the course of method excecution. When the method finishes execution, these variables fall out of scope.
3. Static: - These are the class level variables. They are initialized when the class is loaded in JVM for the first time and remain there as long as the class remains loaded. They are not tied to any particular object instance.
[ Received from Sandesh Sadhale] TOP

**Q: What is the default value of the local variables?**
**A:** The local variables are not initialized to any default value, neither primitives nor object references. If you try to use these variables without initializing them explicitly, the java compiler will not compile the code. It will complain abt the local varaible not being initilized..
[ Received from Sandesh Sadhale] TOP

**Q: How many objects are created in the following piece of code?**
```
MyClass c1, c2, c3;
c1 = new MyClass ();
c3 = new MyClass ();
```
**A:** Only 2 objects are created, c1 and c3. The reference c2 is only declared and not initialized.
[ Received from Sandesh Sadhale] TOP

**Q: Can a public class MyClass be defined in a source file named YourClass.java?**
**A:** No the source file name, if it contains a public class, must be the same as the public class name itself with a .java extension.
[ Received from Sandesh Sadhale] TOP

**Q: Can main method be declared final?**
**A:** Yes, the main method can be declared final, in addition to being public static.
[ Received fromSandesh Sadhale] TOP

**Q: What will be the output of the following statement?**
```
System.out.println ("1" + 3);
```
**A:** It will print 13.
[ Received from Sandesh Sadhale] TOP

**Q: What will be the default values of all the elements of an array defined as an instance variable?**
**A:** If the array is an array of primitive types, then all the elements of the array will be initialized to the default value corresponding to that primitive type. e.g. All the elements of an array of int will be initialized to 0, while that of boolean type will be initialized to false. Whereas if the array is an array of references (of any type), all the elements will be initialized to null.
[ Received from Sandesh Sadhale] TOP