Nexwave
Java Interview
Q & A

What is an Object?

An object is an entity with certain attributes or qualities and behaviors, for a simple example, a 'Laptop' is an object which has certain attributes like weight, color, screen size, manufacturer etc. It has various behaviors or activities to do or act upon, as play games, browse Internet, write/check emails, watch movies, listen music etc. Object is vital of the Software and is combination of two things:

1-State
2-Behavior

State: state is represent his properties
Behavior: Behavior is represent the operation which Object is perform

What is a Class?

A class is a collection of attributes and behaviors of objects with certain similarities and an instance of a class is represented by an object. A simple example of class is a 'Car' which represents variety of Car objects with different attribute values and behaviors. The different objects of 'Car' class can be, for example: A Mercedes Car, a Toyota Car, two different objects from same class but different attributes and different behaviors too.

A class is a blueprint for the particular type of objects. Class describes how the objects of the class looks like and what will be the attributes and the operation a object can do.

What is a method?

A method is the basic unit of functionality contained in a class. It contains the executable body that can be applied to specific object of the class.

The functions and subroutines of a procedural language(like C) are called as methods in an object-oriented language. Like function, a method includes:

- a name
- parameters used to input some values(optional)
- a return type that gives output to another part of the program(atleast void)
- a body of executable code

The following code snippet declares a method whose output (return type) is a double value, the sum of two numbers.

```
public double sum( double a, double b )    {
      Return ( a + b ) ;
}
```

Because methods belong to classes, the above method is to be placed in a class. Methods are not simply called. They must be called with the help of an object. For example:

```
Button btn1 = new Button( ) ;
Button btn2 = new Button( ) ;
btn1.setLabel( “ OK “ ) ;
btn2.setLabel( “ Cancel “ ) ;
```

In the above code, btn1 and btn2 are the objects of Button class. setLabel( ) method of Button class is called with objects btn1 and btn2.

What is OOAD?

Object Oriented Analysis and Design(OOAD) is a methodology to analyze,design and develop application using objects and their relations and message based communication to each other.Everything in OOAD is visualized in terms of objects and classes.OOAD introduced a paradigm shift from thinking and programming procedurally to objects oriented programming.This approach helps in designing complex real time systems with ease.The features like Data Abstraction and Encapsulation, Inheritance and Polymorphism form fundamentals of object oriented programming.

Advantages:

- Enhanced Reusability
- Modular approach towards problem solving which will be
- Better Maintainability
- Better Performance if system is designed cautiously using OOAD concepts

What is Data Abstraction?

Data Abstraction is extraction of essential information for a particular purpose and ignoring the remainder of the information, e.g. a car is consisted of an engine, air filters, a carburetor, a gear box, a steering, a fuel tank, tyres etc. A driver of a car need not to be bothered about several finer points of the car, he/she should know what it requires to drive a car. Take another user, a car mechanic; he will require different set of information in order to repair the car.

What is Data Encapsulation?

Data Encapsulation is wrapping information (attributes and behaviors) within an object. A suitable example is a class as it wraps methods and data within itself. The attributes of a class corresponds to its data members while behavior corresponds to member methods of the class.

What is the difference between Data Abstraction and Information Hiding?

Data Abstraction is often confused with information hiding while they altogether are two different technical concepts. Here are few established definitions of Data Abstraction:

1. "A view of a problem that extracts the essential information relevant to a particular purpose and ignores the remainder of the information." -- [IEEE, 1983]
2. "The essence of abstraction is to extract essential properties while omitting inessential details." -- [Ross et al, 1975]
3. "Abstraction is the selective examination of certain aspects of a problem. The goal of abstraction is to isolate those aspects that are important for some purpose and suppress those aspects that are unimportant." -- [Rumbaugh et al, 1991]
4. "An abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of object and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer." -- [Booch, 1991]
5. While information hiding is not sharing the details of an object with outside world.Here are few standard definitions of Information Hiding which will elaborate more on this concept:
6. "The technique of encapsulating software design decisions in modules in such a way that the module's interfaces reveal little as possible about the module's inner workings; thus each module is a 'black box' to the other modules in the system." -- [IEEE, 1983]
7. "The process of hiding all the details of an object that do not contribute to its essential characteristics; typically, the structure of an object is hidden, as well as the implementation of its methods. The terms information hiding and encapsulation are usually interchangeable." -- [Booch, 1991]
8. "The principle of information hiding is central. It says that modules are used via their specifications, not their implementations. All information about a module, whether concerning data or function, is encapsulated with it and, unless specifically declared public, hidden from other modules." -- [Graham, 1991]

What do you mean by platform independence?

Platform independence means that we can write and compile the java code in one platform (eg Windows)and can execute the class in any other supported platform eg (Linux,Solaris,etc).

What is SDLC? What are the phases of SDLC?

Software Development Life Cycle. The various phases of SDLC are

1. Requirements Gathering: Requirements gathering is done by business analyst. Development team analyse the requirements from the design, architecture & coding perspective.

2. Design: Technical architect works for the high level & low design of the software. Business analyst works for the UI design of the application

3. Coding or development: Development team does the actual coding based on the designed architecture.

4. Testing: In SDLC, actual testing is carried out in this phase. It includes unit testing, integration testing & system testing etc..

5. Deployment: Application is deployed on production environment for real end users.

6. Maintenance: Basically, it includes, post production / deployment support & enhancements.

What is the difference between a JDK and a JVM?
JDK is Java Development Kit which is for development purpose and it includes execution environment also. But JVM is purely a run time environment and hence you will not be able to compile your source files using a JVM.

What is the difference between Requirements & Specifications?
"Requirements" are statements given by the customer as to what needs to be achieved by the software system. Later on these requirements are converted into specifications which are nothing but feasible or implementable requirements.

Whereas "Specifications" are feasible requirements derived from various statements given by the customer. These are the starting point for the product development team.

What is difference between Path and Classpath?
Path and Classpath are operating system level environment variales. Path is used define where the system can find the executables(.exe) files and classpath is used to specify the location .class files.

Should a main() method be compulsorily declared in all java classes?
No not required. main() method should be defined only if the source class is a java application.

Why is the main() method declared static?
main() method is called by the JVM even before the instantiation of the class hence it is declared as static.

Can a main() method be declared final?
Yes. Any inheriting class will not be able to have it's own default main() method.

Does the order of public and static declaration matter in main() method?
No. It doesn't matter but void should always come before main().

Can a source file contain more than one class declaration?
Yes a single source file can contain any number of Class declarations but only one of the class can be declared as public.

Which package is imported by default?
java.lang package is imported by default even without a package declaration.

Can a class be declared as protected?
A class can't be declared as protected. only methods can be declared as protected.

Can you give few examples of final classes defined in Java API?
java.lang.String, java.lang.Math are final classes.

What is use of a abstract variable?
Variables can't be declared as abstract. only classes and methods can be declared as abstract.

Can you create an object of an abstract class?
Not possible. Abstract classes can't be instantiated.

Can a method inside a Interface be declared as final?
No not possible. Doing so will result in compilation error. public and abstract are the only applicable modifiers for method declaration in an interface.

Can an Interface be final?
Not possible. Doing so so will result in compilation error.

Can a class be defined inside an Interface?
Yes it's possible.

Can an Interface be defined inside a class?
Yes it's possible.

What is a Marker Interface?
An Interface which doesn't have any declaration inside but still enforces a mechanism.

Which object oriented Concept is achieved by using overloading and overriding?
Polymorphism.

Why does Java not support operator overloading?
Operator overloading makes the code very difficult to read and maintain. To maintain code simplicity, Java doesn't support operator overloading.

What is Externalizable?
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)

What is Downcasting ?
Downcasting is the casting from a general to a more specific type, i.e. casting down the hierarchy.

What happens to a static variable that is defined within a method of a class ?
Can't do it. You'll get a compilation error.

How many static initializers can you have ?
As many as you want, but the static initializers and class variable initializers are executed in textual order and may not refer to class variables declared
in the class whose declarations appear textually after the use, even though these class variables are in scope.

What is constructor chaining and how is it achieved in Java ?
A child object constructor always first needs to construct its parent (which in turn calls its parent constructor.). In Java it is done via an implicit call to the no-args constructor as the first statement.

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, …).

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

What is the implementation of destroy method in java. Is it native or java code?
This method is not implemented.

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

Is JVM a compiler or an interpreter?
Interpreter.

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 Assertion Error on failure. This keyword should be used for the critical arguments. Meaning, without that the method does nothing.

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

Are JVM's platform independent?

JVM's are not platform independent. JVM's are platform specific run time implementation provided by the vendor.

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.

What are the methods in Object?
clone, equals, wait, finalize, getClass, hashCode, notify, notifyAll, toString.

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

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.

What is the difference between Thread.start() method and Thread.run() method?
Thread.strat() method is used to run the Thread.run() method in a thread. run() method is used to start the thread.

How to know the thread priority?
isAlive() —it returns true when Thread is running
join() these two methods are used to know thread priority.

Why isn't there operator overloading?
Because C++ has proven by example that operator overloading makes code almost impossible to maintain. In fact there very nearly wasn't even method overloading in Java, but it was thought that this was too useful for some very basic methods like print(). Note that some of the classes like DataOutputStream have unoverloaded methods like writeInt() and writeByte().

Does Java have "goto"?
No.

What does the "abstract" keyword mean in front of a method? A class?
Abstract keyword declares either a method or a class. If a method has a abstract keyword in front of it,it is called abstract method.Abstract method hs no body.It has only arguments and return type. Abstract methods act as placeholder methods that are implemented in the subclasses. Abstract classes can't be instantiated.If a class is declared as abstract,no objects of that class can be created. If a class contains any abstract method it must be declared as abstract.

What is RMI?
RMI stands for Remote Method Invocation. Traditional approaches to executing code on other machines across a network have been confusing as well as tedious and error-prone to implement. The nicest way to think about this problem is that some object happens to live on another machine, and that you can send a message to the remote object and get a result as if the object lived on your local machine. This simplification is exactly what Java Remote Method Invocation (RMI) allows you to do.

Can I use <object> instead of <applet>?
Yes! But Sun recommends using <applet>.

Why to use Applet, what type of security it is providing?
When Applet arrives on client's machine, it has limited access to resources, this helps to run Applet successfully without disturbing client`s machine disk.

What is the Java API?
The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets.

What is implicit casting?
Implicit casting is the process of simply assigning one entity to another without any transformation guidance to the compiler. This type of casting is not permitted in all kinds of transformations and may not work for all scenarios.
Example:

```
int i = 1000;
long j = i; //Implicit casting
```

What is explicit casting?
Explicit casting in the process in which the complier are specifically informed to about transforming the object.
Example:

```
long i = 700.20;
int j = (int) i; //Explicit casting
```

In System.out.println(), what is System, out and println?
System is a predefined final class, out is a PrintStream object and println is a built-in overloaded method in the out object.

What is the difference between declaring a variable and defining a variable?
In declaration we only mention the type of the variable and its name without initializing it. Defining means declaration + initialization.

Example:
String s; is just a declaration while String s = new String("bob"); Or String s = "bob"; are both definitions.

How can one prove that the array is not null but empty?
Print array.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 array.length.

What is phantom memory?
Phantom memory is false memory. Memory that does not exist

What is Inheritance and what are different types of it?
Inheritance is a mechanism by which a specific object acquires attributes and behaviors of more general objects. In OOP terminology ,Inheritance is the mechanism which allows a Class 'A' to inherit properties of Class 'B' and we say 'A inherits from B' or in other words B is a 'Superclass' / 'Parent class' while A is a 'Subclass'/'Child class'. A typical example of inheritance is a family tree which consists of son, father, grandfather, great grandfather and so on. The different types of Inheritance are:

1. Single Inheritance
2. Multiple Inheritance
3. Multilevel Inheritance
4. Hierarchical Inheritance
5. Hybrid Inheritance

In single inheritance, a class inherits implementation from only one super class. For example, if class B inherits from class A, class B will acquire all the members declared in class A.

B------>A(Parent)

In multilevel inheritance, a class inherits from a derived class (or subclass). For example, if class C inherits from class B, and class B inherits from class A, class C will acquire all the members declared in class B as well as all the members declared in class A.

C------->B------>A(Parent)

In hierarchical inheritance, many sub classes inherit from a single super class. For example, if classes B, C, and D inherit from class A, classes B, C, and D will acquire all the members declared in class A.

B------>A(Parent)C------>A(Parent)D------>A(Parent)

In multiple inheritance, a class inherits from several super classes. For example, if class C inherits from both class A and class B, class C will acquire all the members declared in class A as well as all the members declared in class B. Multiple inheritance is not directly supported by Java but through Interfaces one can.

C------>A(Parent)C------>B(Parent)

A hybrid inheritance is a combination of any two of the above discussed inheritance types

Why Java uses Singly rooted hierarchy?

All objects in Java are inherited from same base class called 'Object'. In Java all objects have common interface to implement and it makes implementaion of Garbage collector lot easier in Java. The necessary implementaion is provided in base class , and the garbage collector can then send the necessary messages to every object in the system. Without singly rooted hierarchy, it would have been difficult to implement garbage collection feature. It enables lot of ease to programmers not to be bothered about memory management while development. It greatly simplifies argument passing amongst object too on the heap. As Java started from scratch and has no backward compatibility issues with any existing language, it was a logical choice to use the singly-rooted hierarchy in common with most other object-oriented programming languages.

Why does Java not support Multiple Inheritance?

Java does not support multiple inheritance atleast not the way it does in case of C++.In designer's view Multiple Inheritance poses many problems and confusions than it solves.e.g. famous Diamond problem The diamond problem is an ambiguity that can occur when a class multiply inherits from two classes that both descend from a common super class. In such scenarios assuming if Java implements multiple inheritance then it would be difficult to know which method is to be called by an inheriting class object of two of the super classes. In Java, interfaces solve all these ambiguities caused by the diamond problem. Through interfaces, Java allows multiple inheritance of interface but not of implementation. Implementation, which includes instance variables and method implementations, is always singly inherited. As a result, confusion will never arise in Java over which inherited instance variable or method implementation to use.

What is the difference between an instance variable and class variable?

An instance variable represents separate data for each instance (because of separate memory allocation) and a class variable represents a single data shared by whole class(that is, by all instances).

An instance variable represents data for which each instance has its own copy.
A class variable belongs to the class as a whole. All instances access the same single copy. That is, a class variable is shared resource and a means of communication between all instances of a class. class variables declared as **static** keyword.

```
public class Demo   {
      int x ;                // instance variable
      static int y ;         // static variable
}
```

What is the difference between (instance) method and a class method ?

An instance method has a this instance reference and whereas a class method does not. An instance method works always with an instance of a class. An instance method should be called with an instance. A class method can be called without an instance. A class method should be declared with a **static** keyword.

For example, String class has both instance and class methods. length( ) method is an instance method and valueOf( ) is a class(declared with static keyword) method.

```
String str = “ SNRao “ ;
int x = str.length( ) ;

int y = “ 10 “ ;
String s1 = String.valueOf( y ) ;
```

Discuss the restrictions on a static method?

static is a keyword in Java. This keyword can be used with methods and instance variables. Static variable is called as **class variable** as it is shared by all the objects of the class.

1. Generally to call a method from a main( ) method, we require an object of the class. But a static method can be called without an object. The following program illustrates :

```
    public class StaticDemo    {
        public static void display( )    {
                System.out.println ( " Hello1 " ) ;
        }
        public static void main( String args[ ] )    {
                display( ) ;  //  because method is static, it can be called without an
object
        }
        }
```

In the above program, main and display are both static methods. That is calling and called methods are both static.

2. Static method can be called from another static method. A non-static method can be called from a static method. But **a static method cannot call a non-static method**. The following program illustrates :

```
    public class StaticDemo    {
        public  void show( )    {
        System.out.println ( " Hello1 " ) ;
        }
        public  static void display( )    {
        show( ) ;                    // raises an error
        }
        public static void main( String args[ ] )    {
        StaticDemo sd = new StaticDemo( ) ;
        sd.display( ) ;
        }
    }
```

3. Static methods do not have “ **this** “ reference, because static method can be called without an object.
4. Static methods cannot be called with “ **super** “ keyword in inheritance.

What is Early Binding?

The assignment of types to variables and expressions at compilation time is known as 'Early Binding', it is also called 'static binding' and 'static typing'.

What is Late Binding?

When an object is sent a message then it does not know itself what type it is, the runtime environment will decide about function calling over an object. This feature of connecting an object with its associated message at runtime is known as Polymorphism or Late binding or Dynamic binding.

What is method overloading?

A method with changed formal parameters will lead to implementing method overloading.

int calculateSum(int i,int j)
float calculateSum(float i,int j)
double calculateSum(double i,int j)
float calculateSum(int i,float j)

What is method overriding?

The method with same signature but with changed implementation lead to method overriding and that can occur in a parent child relation of classes. A method defined in parent class can be overridden in its child class with different implementation from its base class.

An example: We will define a base class called Circle

```
class Circle {
        //declaring the instance variableprotected double radius;
        public Circle(double radius) {
                this.radius = radius;
        }
        // other method definitions here
        public double getArea(){
                return Math.PI*radius*radius;
        }
```

```
              //this method returns the area of the circle

       }// end of class circle
```

When the getArea method is invoked from an instance of the Circle class, the method returns the area of the circle.

The next step is to define a subclass to override the getArea() method in the Circle class. The derived class will be the Cylinder class. The getArea() method in the Circle class computes the area of a circle, while the getArea method in the Cylinder class computes the surface area of a cylinder.
The Cylinder class is defined below.

```
       class Cylinder extends Circle {
              //declaring the instance variableprotected double length;
              public Cylinder(double radius, double length) {
                     super(radius);this.length = length;
              }
              // other method definitions herepublic
              double getArea(){
                     // method overriden here
                     return 2*super.getArea()+2*Math.PI*radius*length;
              }//this method returns the cylinder surface area

       }// end of class Cylinder
```

When the overriden method (getArea) is invoked for an object of the Cylinder class, the new definition of the method is called and not the old definition from the superclass(Circle).

Differences between Overloading and Overriding:

Overloading	Over-riding
Occurs in the same class	Occurs between two classes known as superclass and subclass.
No method hides another.	Subclass method hides(blocks) superclass mehod
No method replaces the other.	Subclass method replaces the suerclasss method.
Different method signatures	Same method signataure.
May have different return types	Should have the same return type.
May have different declared exceptions	Must have compatible declared exceptions.

How is Java different from C++?

Java is a platform independent, object oriented language while C++ is having some of its features from C, which is a procedural language so it is not pure object oriented. Even Java is not 100% pure object oriented.

1. Pointers are supported in C++ while not in Java. The memory management is done automatically with help of part of JVM called Garbage Collector.
2. Multiple inheritance is not supported in Java but supported in C++.
3. There are no structures and unions in Java.
4. There is no scope resolution operator in Java (::).
5. There are no destructors in Java like C++.
6. There is no virtual keyword in Java because all non-static method use dynamic binding.
7. Java is compiled into byte code and then executed in a virtual machine with automatic memory management. C++ is compiled into native code.

What is the difference between procedural and object-oriented programs?

a) In procedural program, programming logic follows certain procedures and the instructions are executed one after another. In OOP program, unit of program is object, which is nothing but combination of data and code. b) In procedural program, data is exposed to the whole program whereas in OOPs program, it is accessible with in the object and which in turn assures the security of the code.

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.

What is the difference between Assignment and Initialization?
Assignment can be done as many times as desired whereas initialization can be done only once.

What is OOPs?
Object oriented programming organizes a program around its data, i. e. , objects and a set of well defined interfaces to that data. An object-oriented program can be characterized as data controlling access to code.

What are Class, Constructor and Primitive data types?
Class is a template for multiple objects with similar features and it is a blue print for objects. It defines a type of object according to the data the object can hold and the operations the object can perform. Constructor is a special kind of method that determines how an object is initialized when created. Primitive data types are 8 types and they are: byte, short, int, long, float, double, boolean, char.

What is an Object and how do you allocate memory to it?
Object is an instance of a class and it is a software unit that combines a structured set of data with a set of operations for inspecting and manipulating that data. When an object is created using new operator, memory is allocated to it.

What is the difference between constructor and method?
Constructor will be automatically invoked when an object is created whereas method has to be called explicitly.

What are methods and how are they defined?
Methods are functions that operate on instances of classes in which they are defined. Objects can communicate with each other using methods and can call methods in other classes. Method definition has four parts. They are name of the method, type of object or primitive type the method returns, a list of parameters and the body of the method. A method's signature is a combination of the first three parts mentioned above.

What is the use of bin and lib in JDK?
Bin contains all tools such as javac, appletviewer, awt tool, etc., whereas lib contains API and all packages.

What is casting?
Casting is used to convert the value of one type to another.

How many ways can an argument be passed to a subroutine and explain them?
An argument can be passed in two ways. They are passing by value and passing by reference. Passing by value: This method copies the value of an argument into the formal parameter of the subroutine. Passing by reference: In this method, a reference to an argument (not the value of the argument) is passed to the parameter.

What is the difference between an argument and a parameter?
While defining method, variables passed in the method are called parameters. While using those methods, values passed to those variables are called arguments.

What are different types of access modifiers?

- public: Any thing declared as public can be accessed from anywhere.
- private: Any thing declared as private can't be seen outside of its class.
- protected: Any thing declared as protected can be accessed by classes in the same package and subclasses in the other packages.
- default modifier : Can be accessed only to classes in the same package.

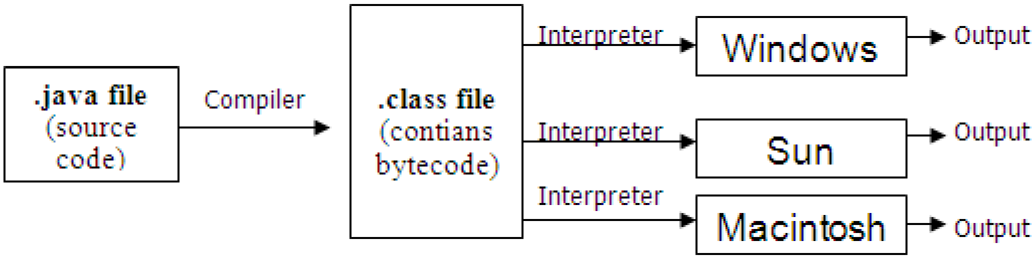
What is final, finalize() and finally?

final : final keyword can be used for class, method and variables. A final class cannot be subclassed and it prevents other programmers from subclassing a secure class to invoke insecure methods. A final method can't be overridden. A final variable can't change from its initialized value.
finalize() : finalize() method is used just before an object is destroyed and can be called just prior to garbage collection.
finally : finally, a key word used in exception handling, creates a block of code that will be executed after a try/catch block has completed and before the code following the try/catch block. The finally block will execute whether or not an exception is thrown. For example, if a method opens a file upon exit, then you will not want the code that closes the file to be bypassed by the exception-handling mechanism. This finally keyword is designed to address this contingency.

What is UNICODE?
Unicode is used for internal representation of characters and strings and it uses 16 bits to represent each other.

Differentiate between Bytecode and Unicode ?
Java introduces two new words to the computer world – Bytecode and Unicode.

Bytecode: When we compile a .java file, we get a .class file. The .class file can run on any operating system irrespective of platform on which it was compiled. For this reason, Java is called platform independent. But the .exe file of C language is not platform independent. .exe file contains binary code. Java's .class file contains bytecode. This bytecode makes Java cross platform. Java compiler produces bytecodes. Any JVM, can run these bytecode and produce output. Bytecode is a machine-independent intermediate language known to a Java interpreter. Java interpreter parses bytecode into an output. That is why Java is said, " write once, run anywhere ".



Unicode:
ASCII(extended) character range is 0 to 255. We cannot add one more character, if we do want. Only English alphabets has got corresponding ASCII values. That is why we cannot write a C program in any other language than English.

Java's motto is internationalization. That is, it supports many world languages, like Telugu, Kannada, Greek, Japanese etc. That is, there is a corresponding ASCII (Unicode) value in Java for all these international languages. This is possible due to the size of character of 2 bytes. That is, the character can represent values ranging from 0 to 65,535. This range is called Unicode. We can say ASCII is a subset of Unicode. Upto 255, Unicode represents ASCII range and afterwards it adds its own values for the alphabets of many world languages. Unicode is already includes up to 34,128 characters.

What is Garbage Collection and how to call it explicitly?
When an object is no longer referred to by any variable, java automatically reclaims memory used by that object. This is known as garbage collection. System.gc() method may be used to call it explicitly.

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

How you can force the garbage collection?
Garbage collection automatic process and can't be forced. You could request it by calling System.gc(). JVM does not guarantee that GC will be started immediately.

What is finalize() method?
finalize () method is used just before an object is destroyed and can be called just prior to garbage collection.

What are Transient and Volatile Modifiers?
Transient: The transient modifier applies to variables only and it is not stored as part of its object's Persistent state. Transient variables are not serialized. Volatile: Volatile modifier applies to variables only and it tells the compiler that the variable modified by volatile can be changed unexpectedly by other parts of the program.

What is method overloading and method overriding?
Method overloading: When a method in a class having the same method name with different arguments is said to be method overloading. Method overriding: When a method in a class having the same method name with same arguments is said to be method overriding.

What is difference between overloading and overriding?
a) In overloading, there is a relationship between methods available in the same class whereas in overriding, there is relationship between a superclass method and subclass method. b) Overloading does not block inheritance from the superclass whereas overriding blocks inheritance from the superclass. c) In overloading, separate methods share the same name whereas in overriding, subclass method replaces the superclass. d) Overloading must have different method signatures whereas overriding must have same signature.

What is meant by Inheritance and what are its advantages?
Inheritance is the process of inheriting all the features from a class. The advantages of inheritance are reusability of code and accessibility of variables and methods of the super class by subclasses.

What is the difference between this() and super()?
this() can be used to invoke a constructor of the same class whereas super() can be used to invoke a super class constructor.

What is the difference between superclass and subclass?
A super class is a class that is inherited whereas sub class is a class that does the inheriting.

What modifiers may be used with top-level class?
public, abstract and final can be used for top-level class.

What are inner class and anonymous class?
Inner class: classes defined in other classes, including those defined in methods are called inner classes. An inner class can have any accessibility including private.
Anonymous class: Anonymous class is a class defined inside a method without a name and is instantiated and declared in the same place and cannot have explicit constructors.

What is a package?
A package is a collection of classes and interfaces that provides a high-level layer of access protection and name space management.

What is a reflection package?
java. lang. reflect package has the ability to analyze itself in runtime.

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:

a)Declaring methods that one or more classes are expected to implement
b)Capturing similarities between unrelated classes without forcing a class relationship.
c)Determining an object's programming interface without revealing the actual body of the class.

What is an abstract class?

An abstract class is a class designed with implementation gaps for subclasses to fill in and is deliberately incomplete.

What is the difference between Integer and int?
a) Integer is a class defined in the java. lang package, whereas int is a primitive data type defined in the Java language itself. Java does not automatically convert from one to the other. b) Integer can be used as an argument for a method that requires an object, whereas int can be used for calculations.

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.

What is the difference between abstract class and interface?
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.

Differences between an abstract class and interface:

	Abstract class	**Interface**
1.	Abstract class may contain all concrete methods or abstract methods or a mixture.	Interface should contain only abstract methods.
2.	The methods may have any access specifier except private.	The access specifier must be public.
3.	The variables can have any access specifier except private.	The variables must be public, static and final.
4.	Multiple inheritance is not possible.	Multiple inheritance is possible.
5.	Inheritance goes with extends keyword.	Inheritance goes with implements keyword.
6.	abstract keyword must be included in method declaration.	public and abstract, even if omitted, they are assumed by default.

The following is a typical example for an interface defined in **java.awt.event** package:

```
public interface WindowListener extends EventListener  {
      public void windowClosing( WindowEvent e )  ;
      public void windowClosed( WindowEvent e )  ;
      public void windowOpened( WindowEvent e )  ;
      public void windowActivated( WindowEvent e )  ;
      public void windowDeactivated( WindowEvent e )  ;
      public void windowIconified( WindowEvent e )  ;
      public void windowDeiconified( WindowEvent e )  ;
}
```

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.

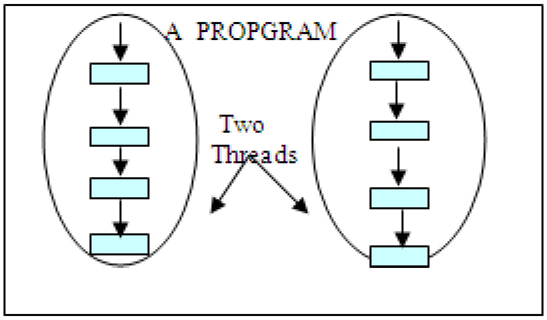
What is the difference between String and String Buffer?
a) String objects are constants and immutable whereas StringBuffer objects are not. b) String class supports constant strings whereas StringBuffer class supports growable and modifiable strings.

What is the difference between Array and vector?
Array is a set of related data type and static whereas vector is a growable array of objects and dynamic.

What is the difference between exception and error?
The exception class defines mild error conditions that your program encounters. Exceptions can occur when trying to open the file, which does not exist, the network connection is disrupted, operands being manipulated are out of prescribed ranges, the class file you are interested in loading is missing. The error class defines serious error conditions that you should not attempt to recover from. In most cases it is advisable to let the program terminate when such an error is encountered.

What is the difference between process and thread?
Process is a program in execution whereas thread is a separate path of execution in a program.

What is a thread?
A thread is a single sequential (separate) flow of control within program. Sometimes, it is called an execution context or light weight process. A thread itself is not a program. A thread cannot run on its own(as it is a part of a program). Rather, it runs within a program. A program can be divided into a number of packets of code ---- each representing a thread having its own separate flow of control.



Two threads running concurrently in a single program

Light weight process: A thread is considered a light weight process because it runs within the context of a program and takes advantage of the resources allocated to that program.
Heavy weight process: In the heavy weight process, the control changes in between threads belonging to different processes. (In light weight process, the control changes in between threads belonging to same(one) process).
Execution contest: A thread will have its own execution stack and program counter. The code running within the thread works only within that context.

One of the strengths of Java is its support for multithreading. All the classes needed to write a multithreaded program are included in the default imported package java.lang through class Object, class Thread and interface Runnable.

How to create a thread and start running?
We can create a thread by instantiating Thread class(or of its subclasses) and invoke start( ) on it.

A thread is an instance of Thread class. To create a thread of a class, we must extend the class with Thread class or Runnable interface.

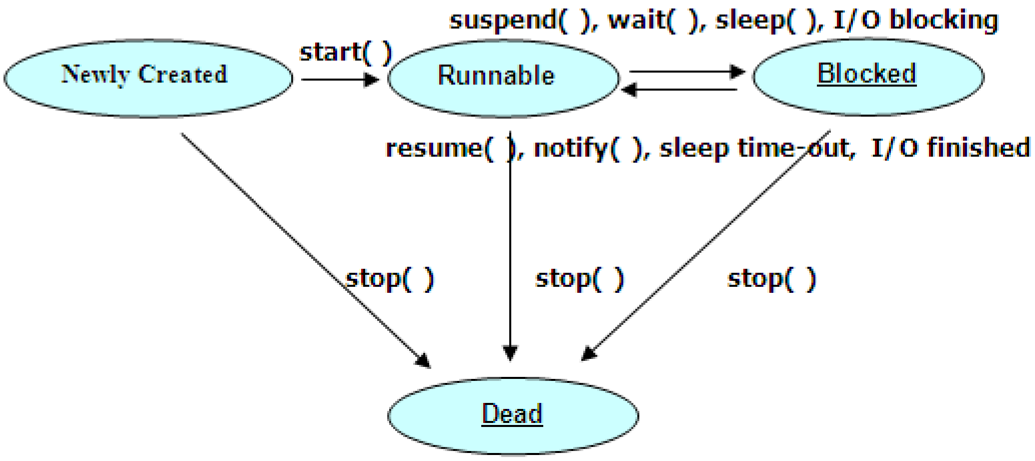
```
public class Demo  extends Thread   {
        public void run( )   {
                for( int i = 0 ; i < 10 ; i + + )
                        System.out.println( i ) ;  // prints 0 to 9 numbers
        }
        public static void main( String args [ ] )   {
                Demo d1 = new Demo( ) ;
                d1.start( ) ; // without this statement, run( ) method is not called
        }
}
```

In the above program, object d1, implicitly becomes a thread object(or to say, it can make use of all the methods of Thread class). That is thread d1 is created. d1.start( ) makes thread to start functioning by executing the code of run( ) method.

What is multithreading and what are the methods for inter-thread communication and what is the class in which these methods are defined?
Multithreading is the mechanism in which more than one thread run independent of each other within the process. wait (), notify () and notifyAll() methods can be used for inter-thread communication and these methods are in Object class. wait() : When a thread executes a call to wait() method, it surrenders the object lock and enters into a waiting state. notify() or notifyAll() : To remove a thread from the waiting state, some other thread must make a call to notify() or notifyAll() method on the same object.

What is the class and interface in java to create thread and which is the most advantageous method? Thread class and Runnable interface can be used to create threads and using Runnable interface is the most advantageous method to create threads because we need not extend thread class here.

Discuss the life cycle of a thread?
Just like an applet and servlet, threads too got a life cycle between their birth and death.


suspend( ), wait( ), sleep( ), I/O blocking
Newly Created
start( )
Runnable
Blocked
resume( ), notify( ), sleep time-out, I/O finished
stop( )
stop( )
stop( )
Dead

The Thread life Cycle

Born state: When a thread is created, the thread will be in born state. The thread in born state is not eligible for processor time.

```
        Thread t1 = new Thread( ) ; // thread t1 is created and is in born state.
```

Runnable state: After the creation of thread, when start( ) method is called, it comes into runnable state. The thread in runnable state is eligible for processor time. In the runnable state, the thread calls run( ) method and executes the code.

```
                  t1.start( ) ;  This method gets the thread into runnable state.
```

Blocked state: We can block the microprocessor time for a thread by calling sleep(), wait() or suspend() methods on it. The thread in blocked state is alive and is not eligible for processor time even if the processor is idle.

Dead state: The thread in dead state is liable for garbage collection.
Events when a thread can come into dead state:

1. When stop( ) method is called on the thread.
2. When the thread completes the execution of run( ) method.
3. When the power supply to the system is cut off.

What are the states associated in the thread?
Thread contains ready, running, waiting and dead states.

What is synchronization?
Synchronization is the mechanism that ensures that only one thread is accessed the resources at a time.

When you will synchronize a piece of your code?
When you expect your code will be accessed by different threads and these threads may change a particular data causing data corruption.

What is deadlock?
When two threads are waiting each other and can't precede the program is said to be deadlock.

What is daemon thread and which method is used to create the daemon thread?
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.

Are there any global variables in Java, which can be accessed by other part of your program?

No, it is not the main method in which you define variables. Global variables is not possible because concept of encapsulation is eliminated here.

What is an applet?
Applet is a dynamic and interactive program that runs inside a web page displayed by a java capable browser.

What is the difference between applications and applets?

a) Application must be run on local machine whereas applet needs no explicit installation on local machine.
b) Application must be run explicitly within a java-compatible virtual machine whereas applet loads and runs itself automatically in a java-enabled browser.
c) Application starts execution with its main method whereas applet starts execution with its init method.
d) Application can run with or without graphical user interface whereas applet must run within a graphical user interface.

How does applet recognize the height and width?
Using getParameters() method.

When do you use codebase in applet?
When the applet class file is not in the same directory, codebase is used.

What is the lifecycle of an applet?
init() method - Can be called when an applet is first loaded start() method - Can be called each time an applet is started. paint() method - Can be called when the applet is minimized or maximized. stop() method - Can be used when the browser moves off the applet's page. destroy() method - Can be called when the browser is finished with the applet.

How do you set security in applets
using setSecurityManager() method

What is an event and what are the models available for event handling?
An event is an event object that describes a state of change in a source. In other words, event occurs when an action is generated, like pressing button, clicking mouse, selecting a list, etc. There are two types of models for handling events and they are: a) event-inheritance model and b) event-delegation model

What are the advantages of the model over the event-inheritance model?
The event-delegation model has two advantages over the event-inheritance model. They are: a)It enables event handling by objects other than the ones that generate the events. This allows a clean separation between a component's design and its use. b)It performs much better in applications where many events are generated. This performance improvement is due to the fact that the event-delegation model does not have to be repeatedly process unhandled events as is the case of the event-inheritance.

What is source and listener?
source: A source is an object that generates an event. This occurs when the internal state of that object changes in some way.
listener: A listener is an object that is notified when an event occurs. It has two major requirements. First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications.

What is adapter class?
An adapter class provides an empty implementation of all methods in an event listener interface. Adapter classes are useful when you want to receive and process only some of the events that are handled by a particular event listener interface. You can define a new class to act listener by extending one of the adapter classes and implementing only those events in which you are interested. For example, the MouseMotionAdapter class has two methods, mouseDragged()and mouseMoved(). The signatures of these empty are exactly as defined in the MouseMotionListener interface. If you are interested in only mouse drag events, then you could simply extend MouseMotionAdapter and implement mouseDragged() .

What is meant by controls and what are different types of controls in AWT?
Controls are components that allow a user to interact with your application and the AWT supports the following types of controls: Labels, Push Buttons, Check Boxes, Choice Lists, Lists, Scrollbars, Text Components. These controls are subclasses of Component.

What is the difference between choice and list?
A Choice is displayed in a compact form that requires you to pull it down to see the list of available choices and only one item may be selected from a choice. A List may be displayed in such a way that several list items are visible and it supports the selection of one or more list items.

What is the difference between scrollbar and scrollpane?
A Scrollbar is a Component, but not a Container whereas Scrollpane is a Conatiner and handles its own events and perform its own scrolling.

What is a layout manager and what are different types of layout managers available in java AWT?
A layout manager is an object that is used to organize components in a container. The different layouts are available are FlowLayout, BorderLayout, CardLayout, GridLayout and GridBagLayout.

How are the elements of different layouts organized?
FlowLayout: The elements of a FlowLayout are organized in a top to bottom, left to right fashion. BorderLayout: The elements of a BorderLayout are organized at the borders (North, South, East and West) and the center of a container. CardLayout: The elements of a CardLayout are stacked, on top of the other, like a deck of cards. GridLayout: The elements of a GridLayout are of equal size and are laid out using the square of a grid. GridBagLayout: The elements of a GridBagLayout are organized according to a grid. However, the elements are of different size and may occupy more than one row or column of the grid. In addition, the rows and columns may have different sizes.

Which containers use a Border layout as their default layout?
Window, Frame and Dialog classes use a BorderLayout as their layout.

Which containers use a Flow layout as their default layout?
Panel and Applet classes use the FlowLayout as their default layout.

What are wrapper classes? How to convert primitive data types into Objects ?
Wrapper classes are classes that allow primitive types to be accessed as objects. We can use these objects where primitive data types are required as objects.

One circumstance where we require primitive data types as objects is when we want to store them in data structures of Java like Vector or Stack.

Every primitive data type has got a corresponding wrapper class with which we can convert it into an object. For example, int has got Integer class, double has got Double class and boolean has got Boolean class.

The following code converts int into Integer object.

```
        int x = 10 ;
        Integer i1 = new Integer( x ) ;//We say object i1 wraps primitive data type int x.
```

The following code converts double into Double object.

```
        double x = 10.5 ;
        Double d1 = new Double( x ) ; //We say object d1 wraps primitive data type double.
```

The wrapper classes also provide methods to retrieve primitive data type values from objects.

```
        int y = i1.intValue( ) ;          // to get back int value from i1 object
        double y = d1.doubleValue( ) ;    // to get back double value from d1 object
```

Discuss the rules governing type cast operation on primitive data types?

There are 8 primitive data types in Java – byte, short, int, long, float, double, char and boolean. One type can be converted into another implicitly(by the system) or explicitly(by the programmer). The following rules apply.

1. The data types that occupies less memory can be assigned with to data types that occupies more memory. The following casting is performed imlicitly.

```
            int x = 5 ;
            double y = x ;
            System.out.println( y ) ;      // prints 5.0
```

In the above code, x (int occupies 4 bytes) can be assigned to d (double occupies 8 bytes) and this assignment is done implicitly(automatically) without an extra code by the programmer.

2. But the following code does not compile as explicit casting is needed by the programmer.

```
            double d = 5.6 ;
            int i = d ;
            System.out.println( i ) ;     // raises a compilation error
```

In the above program, d (of 8 bytes) can be assigned to i (of 4 bytes). It needs explicit casting. The above code re-written as follows :

```
            double d = 5.6 ;
            int i = ( int ) d ;
            System.out.println( i ) ;      // prints 5
```

This explicit casting sometimes results in data truncation and precision of the value is lost.

3. An important point to note is boolean data type cannot be cast to any other data type. It is incompatible with any other data type. Always conversion(either explicitly or implictly) is possible between compatible data types only. The following code raises a compilation error :

```
    boolean b = true ;
    int x = b ;
```

What is object casting and explain it’s rules ? or Can we cast in between objects ?

Truly speaking, we can not cast in between objects even though all objects in Java take 32-bits of memory. But, it is possible to cast in between a subclass object and superclass object.

Object casting is possible in between the classes involved in inheritance only.

```
    class Flower    {    }
        class Rose      {    }              // observe there is no extends keyword here
        Flower f = new Flower( ) ;
        Rose r = new Rose( ) ;
```

The following statements causes compilation error because objects a and b are not connected at all together.

```
        f = r ;
        r = f ;
```

But the following code does not cause compilation error as class Flower and Class Rose are involved in inheritance.

Rules of casting:

1. a subclass object can be assigned to a superclass object(implicit object conversion takes place).
2. A superclass object cannot be assigned to a subclass object and it needs explicit object casting.

```
        class Flower    {    }
        class Rose extends Flower      {    } // observe extends keyword here
        Flower f = new Flower( ) ;
        Rose r = new Rose( ) ;

        f = r ;      // it is legal, as per rule 1.
        r = f ;      // it is illegal, as per rule 2
        r = ( Rose ) f ;  // now legal, as explicit casing is done
```

What are Vector, Hashtable, LinkedList and Enumeration?

Vector : The Vector class provides the capability to implement a growable array of objects. Hashtable : The Hashtable class implements a Hashtable data structure. A Hashtable indexes and stores objects in a dictionary using hash codes as the object's keys. Hash codes are integer values that identify objects. LinkedList: Removing or inserting elements in the middle of an array can be done using LinkedList. A LinkedList stores each object in a separate link whereas an array stores object references in consecutive locations. Enumeration: An object that implements the Enumeration interface generates a series of elements, one at a time. It has two methods, namely hasMoreElements() and nextElement(). HasMoreElemnts() tests if this enumeration has more elements and nextElement method returns successive elements of the series.

What is the difference between set and list?
Set stores elements in an unordered way but does not contain duplicate elements, whereas list stores elements in an ordered way but may contain duplicate elements.

What is a stream and what are the types of Streams and classes of the Streams?
A Stream is an abstraction that either produces or consumes information. There are two types of Streams and they are: **Byte Streams**: Provide a convenient means for handling input and output of bytes. **Character Streams:** Provide a convenient means for handling input & output of characters.
Byte Streams classes: Are defined by using two abstract classes, namely InputStream and OutputStream. Character Streams classes: Are defined by using two abstract classes, namely Reader and Writer.

What is the difference between Reader/Writer and InputStream/Output Stream?
The Reader/Writer class is character-oriented and the InputStream/OutputStream class is byte-oriented.

What is an I/O filter?
An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another.

What is serialization and deserialization?
Serialization is the process of writing the state of an object to a byte stream. Deserialization is the process of restoring these objects.

What is a transient variable?
transient variable is a variable that may not be serialized.

Which containers use a border Layout as their default layout?
The window, Frame and Dialog classes use a border layout as their default layout.

Why do threads block on I/O?
Threads block on i/o (that is enters the waiting state) so that other threads may execute while the i/o Operation is performed.

How are Observer and Observable used?
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.

What is synchronization and why is it important?
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.

Can a lock be acquired on a class?
Yes, a lock can be acquired on a class. This lock is acquired on the class's Class object.

What's new with the stop(), suspend() and resume() methods in JDK 1.2?
The stop(), suspend() and resume() methods have been deprecated in JDK 1.2.

Is null a keyword?
The null value is not a keyword.

What is the preferred size of a component?
The preferred size of a component is the minimum component size that will allow the component to display normally.

What method is used to specify a container's layout?
The setLayout() method is used to specify a container's layout.

Which containers use a FlowLayout as their default layout?
The Panel and Applet classes use the FlowLayout as their default layout.

What state does a thread enter when it terminates its processing?
When a thread terminates its processing, it enters the dead state.

What is the Collections API?
The Collections API is a set of classes and interfaces that support operations on collections of objects.

Which characters may be used as the second character of an identifier, but not as the first character of an identifier?
The digits 0 through 9 may not be used as the first character of an identifier but they may be used after the first character of an identifier.

What is the List interface?
The List interface provides support for ordered collections of objects.

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

What is the Vector class?
The Vector class provides the capability to implement a growable array of objects

What modifiers may be used with an inner class that is a member of an outer class?
A (non-local) inner class may be declared as public, protected, private, static, final, or abstract.

What is an Iterator interface?
The Iterator interface is used to step through the elements of a Collection.

What is the difference between the >> and >>> operators?
The >> operator carries the sign bit when shifting right. The >>> zero-fills bits that have been shifted out.

Which method of the Component class is used to set the position and size of a component?
setBounds()

How many bits are used to represent Unicode, ASCII, UTF-16, and UTF-8 characters?
Unicode requires 16 bits and ASCII require 7 bits. Although the ASCII character set uses only 7 bits, it is usually represented as 8 bits. UTF-8 represents characters using 8, 16, and 18 bit patterns. UTF-16 uses 16-bit and larger bit patterns.

What is the difference between yielding and sleeping?
When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.

Which java.util classes and interfaces support event handling?
The EventObject class and the EventListener interface support event processing.

Is sizeof a keyword?
The sizeof operator is not a keyword.

What are wrapped classes?
Wrapped classes are classes that allow primitive types to be accessed as objects.

What restrictions are placed on the location of a package statement within a source code file?
A package statement must appear as the first line in a source code file (excluding blank lines and comments).

Can an object's finalize() method be invoked while it is reachable?
An object's finalize() method cannot be invoked by the garbage collector while the object is still reachable. However, an object's finalize() method may be invoked by other objects.

What is the immediate superclass of the Applet class?
Panel

What is the difference between preemptive scheduling and time slicing?
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.

Name three Component subclasses that support painting.
The Canvas, Frame, Panel, and Applet classes support painting.

What value does readLine() return when it has reached the end of a file?
The readLine() method returns null when it has reached the end of a file.

What is the immediate superclass of the Dialog class?
Window

What is clipping?
Clipping is the process of confining paint operations to a limited area or shape.

What is a native method?
A native method is a method that is implemented in a language other than Java.

Can a for statement loop indefinitely?
Yes, a for statement can loop indefinitely. For example, consider the following:
for(;;) ;

What are order of precedence and associativity, and how are they used?
Order of precedence determines the order in which operators are evaluated in expressions. Associatity determines whether an expression is evaluated left-to-right or right-to-left

When a thread blocks on I/O, what state does it enter?
A thread enters the waiting state when it blocks on I/O.

To what value is a variable of the String type automatically initialized?
The default value of an String type is null.

What is the catch or declare rule for method declarations?
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.

What is the difference between a MenuItem and a CheckboxMenuItem?
The CheckboxMenuItem class extends the MenuItem class to support a menu item that may be checked or unchecked.

What is a task's priority and how is it used in scheduling?
A task's priority is an integer value that identifies the relative order in which it should be executed with respect to other tasks. The scheduler attempts to schedule higher priority tasks before lower priority tasks.

What class is the top of the AWT event hierarchy?
The java.awt.AWTEvent class is the highest-level class in the AWT event-class hierarchy.

When a thread is created and started, what is its initial state?
A thread is in the ready state after it has been created and started.

Can an anonymous class be declared as implementing an interface and extending a class?
An anonymous class may implement an interface or extend a superclass, but may not be declared to do both.

What is the range of the short type?
The range of the short type is -(2^15) to 2^15 - 1.

What is the range of the char type?
The range of the char type is 0 to 2^16 - 1.

In which package are most of the AWT events that support the event-delegation model defined?
Most of the AWT-related events of the event-delegation model are defined in the java.awt.event package. The AWTEvent class is defined in the java.awt package.

What is the immediate superclass of Menu?
MenuItem

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

Which class is the immediate superclass of the MenuComponent class.
Object

What invokes a thread's run() method?
After a thread is started, via its start() method or that of the Thread class, the JVM invokes the thread's run() method when the thread is initially executed.

What is the difference between the Boolean & operator and the && operator?
If an expression involving the Boolean & operator is evaluated, both operands are evaluated. Then the & operator is applied to the operand. When an expression involving the && operator is evaluated, the first operand is evaluated. If the first operand returns a value of true then the second operand is evaluated. The && operator is then applied to the first and second operands. If the first operand evaluates to false, the evaluation of the second operand is skipped.

Name three subclasses of the Component class.
Box.Filler, Button, Canvas, Checkbox, Choice, Container, Label, List, Scrollbar, or TextComponent

What is the GregorianCalendar class?
The GregorianCalendar provides support for traditional Western calendars.

Which Container method is used to cause a container to be laid out and redisplayed?
validate()

What is the purpose of the Runtime class?
The purpose of the Runtime class is to provide access to the Java runtime system.

How many times may an object's finalize() method be invoked by the garbage collector?

An object's finalize() method may only be invoked once by the garbage collector.

What is the purpose of the finally clause of a try-catch-finally statement?
The finally clause is used to provide the capability to execute code no matter whether or not an exception is thrown or caught.

What is the argument type of a program's main() method?
A program's main() method takes an argument of the String[] type.

Which Java operator is right associative?
The = operator is right associative.

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

Can a double value be cast to a byte?
Yes, a double value can be cast to a byte.

What is the difference between a break statement and a continue statement?
A break statement results in the termination of the statement to which it applies (switch, for, do, or while). A continue statement is used to end the current loop iteration and return control to the loop statement.

What must a class do to implement an interface?
It must provide all of the methods in the interface and identify the interface in its implements clause.

What method is invoked to cause an object to begin executing as a separate thread?
The start() method of the Thread class is invoked to cause an object to begin executing as a separate thread.

Name two subclasses of the TextComponent class.
TextField and TextArea

What is the advantage of the event-delegation model over the earlier event-inheritance model?
The event-delegation model has two advantages over the event-inheritance model. First, it enables event handling to be handled by objects other than the ones that generate the events (or their containers). This allows a clean separation between a component's design and its use. The other advantage of the event-delegation model is that it performs much better in applications where many events are generated. This performance improvement is due to the fact that the event-delegation model does not have to repeatedly process unhandled events, as is the case of the event-inheritance model.

Which containers may have a MenuBar?
Frame

How are commas used in the intialization and iterationparts of a for statement?
Commas are used to separate multiple statements within the initialization and iteration parts of a for statement.

What is the purpose of the wait(), notify(), and notifyAll() methods?
The wait(),notify(), and notifyAll() methods are used to provide an efficient way for threads to wait for a shared resource. When a thread executes an object's wait() method, it enters the waiting state. It only enters the ready state after another thread invokes the object's notify() or notifyAll() methods.

What is an abstract method?
An abstract method is a method whose implementation is deferred to a subclass. An abstract method does not contain executable body.

An abstract method declares method name, parameter types and return type and not body(even empty braces constitutes a body, but an empty body). That is functionality is not written. An abstract method is declared with an abstract keyword and ends with a semicolon instead of a method body.

An abstract class allows us to design a new class. A class with abstract methods should be abstract. **Graphics** class defined in **java.awt** package contains several abstract methods of which few are given below:

```
public abstract class Graphics extends Object   :
        public void abstract drawLine( int x1, int y1, int x2, int y2 ) ;
        public void abstract drawRect( int x, int y, int width, int height ) ;
}
```

What is an abstract class ?

An abstract is a class designed with implementations gaps for subclasses to fill in.

An abstract class is an incomplete class. It gives skeleton of methods for the subclasses to implement(to give body). The abstract class should be declared with **abstract keyword**. The subclass should implement all the abstract methods of an interface and even if one is omitted it does not compile. An abstract class can contain 1. all abstract methods 2. all concrete methods 3. a mixture of abstract and concrete methods.

Graphics class defined in **java.awt** package contains several abstract methods of which few are given below:

```
public abstract class Graphics extends Object   :
        public void abstract drawLine( int x1, int y1, int x2, int y2 ) ;
        public void abstract drawRect( int x, int y, int width, int height ) ;

}
```

How are Java source code files named?
A Java source code file takes the name of a public class or interface that is defined within the file. A source code file may contain at most one public class or interface. If a public class or interface is defined within a source code file, then the source code file must take the name of the public class or interface. If no public class or interface is defined within a source code file, then the file must take on a name that is different than its classes and interfaces. Source code files use the .java extension.

What is the relationship between the Canvas class and the Graphics class?
A Canvas object provides access to a Graphics object via its paint() method.

What are the high-level thread states?
The high-level thread states are ready, running, waiting, and dead.

What value does read() return when it has reached the end of a file?
The read() method returns -1 when it has reached the end of a file.

Can a Byte object be cast to a double value?
No, an object cannot be cast to a primitive value.

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.

What is the difference between the String and StringBuffer classes?
String objects are constants. StringBuffer objects are not.

If a variable is declared as private, where may the variable be accessed?
A private variable may only be accessed within the class in which it is declared.

What is an object's lock and which object's have locks?
An object's lock is a mechanism that is used by multiple threads to obtain synchronized access to the object. A thread may execute a synchronized method of an object only after it has acquired the object's lock. All objects and classes have locks. A class's lock is acquired on the class's Class object.

What is the Dictionary class?
The Dictionary class provides the capability to store key-value pairs.

How are the elements of a BorderLayout organized?
The elements of a BorderLayout are organized at the borders (North, South, East, and West) and the center of a container.

What is the % operator?
It is referred to as the modulo or remainder operator. It returns the remainder of dividing the first operand by the second operand.

When can an object reference be cast to an interface reference?
An object reference be cast to an interface reference when the object implements the referenced interface.

What is the difference between a Window and a Frame?
The Frame class extends Window to define a main application window that can have a menu bar.

Which class is extended by all other classes?
The Object class is extended by all other classes.

Can an object be garbage collected while it is still reachable?
A reachable object cannot be garbage collected. Only unreachable objects may be garbage collected..

Is the ternary operator written x : y ? z or x ? y : z ?
It is written x ? y : z.

What is the difference between the Font and FontMetrics classes?
The FontMetrics class is used to define implementation-specific properties, such as ascent and descent, of a Font object.

How is rounding performed under integer division?
The fractional part of the result is truncated. This is known as rounding toward zero.

What happens when a thread cannot acquire a lock on an object?
If a thread attempts to execute a synchronized method or synchronized statement and is unable to acquire an object's lock, it enters the waiting state until the lock becomes available.

What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?
The Reader/Writer class hierarchy is character-oriented, and the InputStream/OutputStream class hierarchy is byte-oriented.

What classes of exceptions may be caught by a catch clause?
A catch clause can catch any exception that may be assigned to the Throwable type. This includes the Error and Exception types.

If a class is declared without any access modifiers, where may the class be accessed?
A class that is declared without any access modifiers is said to have package access. This means that the class can only be accessed by other classes and interfaces that are defined within the same package.

What is the SimpleTimeZone class?
The SimpleTimeZone class provides support for a Gregorian calendar.

What is the Map interface?
The Map interface replaces the JDK 1.1 Dictionary class and is used associate keys with values.

Does a class inherit the constructors of its superclass?

A class does not inherit constructors from any of its superclasses.

For which statements does it make sense to use a label?
The only statements for which it makes sense to use a label are those statements that can enclose a break or continue statement.

What is the purpose of the System class?
The purpose of the System class is to provide access to system resources.

Which TextComponent method is used to set a TextComponent to the read-only state?
setEditable()

How are the elements of a CardLayout organized?
The elements of a CardLayout are stacked, one on top of the other, like a deck of cards.

Is &&= a valid Java operator?
No, it is not.

Name the eight primitive Java types.
The eight primitive types are byte, char, short, int, long, float, double, and boolean.

Which class should you use to obtain design information about an object?
The Class class is used to obtain information about an object's design.

What is the relationship between clipping and repainting?
When a window is repainted by the AWT painting thread, it sets the clipping regions to the area of the window that requires repainting.

Is "abc" a primitive value?
The String literal "abc" is not a primitive value. It is a String object.

What is the relationship between an event-listener interface and an event-adapter class?
An event-listener interface defines the methods that must be implemented by an event handler for a particular kind of event. An event adapter provides a default implementation of an event-listener interface.

What restrictions are placed on the values of each case of a switch statement?
During compilation, the values of each case of a switch statement must evaluate to a value that can be promoted to an int value.

What modifiers may be used with an interface declaration?
An interface may be declared as public or abstract.

Is a class a subclass of itself?
A class is a subclass of itself.

What is the highest-level event class of the event-delegation model?
The java.util.EventObject class is the highest-level class in the event-delegation class hierarchy.

What event results from the clicking of a button?
The ActionEvent event is generated as the result of the clicking of a button.

How can a GUI component handle its own events?
A component can handle its own events by implementing the required event-listener interface and adding itself as its own event listener.

What is the difference between a while statement and a dostatement?

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.

How are the elements of a GridBagLayout organized?
The elements of a GridBagLayout are organized according to a grid. However, the elements are of different sizes and may occupy more than one row or column of the grid. In addition, the rows and columns may have different sizes.

What advantage do Java's layout managers provide over traditional windowing systems?
Java uses layout managers to lay out components in a consistent manner across all windowing platforms. Since Java's layout managers aren't tied to absolute sizing and positioning, they are able to accomodate platform-specific differences among windowing systems.

What is the Collection interface?
The Collection interface provides support for the implementation of a mathematical bag - an unordered collection of objects that may contain duplicates.

What modifiers can be used with a local inner class?
A local inner class may be final or abstract.

What is the difference between static and non-static variables?
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.

What is the difference between the paint() and repaint() methods?
The paint() method supports painting via a Graphics object. The repaint() method is used to cause paint() to be invoked by the AWT painting thread.

What is the purpose of the File class?
The File class is used to create objects that provide access to the files and directories of a local file system.

Can an exception be rethrown?
Yes, an exception can be rethrown.

Which Math method is used to calculate the absolute value of a number?
The abs() method is used to calculate absolute values.

How does multithreading take place on a computer with a single CPU?
The operating system's task scheduler allocates execution time to multiple tasks. By quickly switching between executing tasks, it creates the impression that tasks execute sequentially.

When does the compiler supply a default constructor for a class?
The compiler supplies a default constructor for a class if no other constructors are provided.

When is the finally clause of a try-catch-finally statement executed?
The finally clause of the try-catch-finally statement is always executed unless the thread of execution terminates or an exception occurs within the execution of the finally clause.

Which class is the immediate superclass of the Container class?
Component

If a method is declared as protected, where may the method be accessed?
A protected method may only be accessed by classes or interfaces of the same package or by subclasses of the class in which it is declared.

How can the Checkbox class be used to create a radio button?

By associating Checkbox objects with a CheckboxGroup.

Which non-Unicode letter characters may be used as the first character of an identifier?
The non-Unicode letter characters $ and _ may appear as the first character of an identifier

What restrictions are placed on method overloading?
Two methods may not have the same name and argument list but different return types.

What happens when you invoke a thread's interrupt method while it is sleeping or waiting?
When a task's interrupt() method is executed, the task enters the ready state. The next time the task enters the running state, an InterruptedException is thrown.

What is casting?
There are two types of casting, casting between primitive numeric types and casting between object references. Casting between numeric types is used to convert larger values, such as double values, to smaller values, such as byte values. Casting between object references is used to refer to an object by a compatible class, interface, or array type reference.

What is the return type of a program's main() method?
A program's main() method has a void return type.

Name four Container classes.
Window, Frame, Dialog, FileDialog, Panel, Applet, or ScrollPane

What is the difference between a Choice and a List?
A Choice is displayed in a compact form that requires you to pull it down to see the list of available choices. Only one item may be selected from a Choice. A List may be displayed in such a way that several List items are visible. A List supports the selection of one or more List items.

What class of exceptions are generated by the Java run-time system?
The Java runtime system generates RuntimeException and Error exceptions.

What class allows you to read objects directly from a stream?
The ObjectInputStream class supports the reading of objects from input streams.

What is the difference between a field variable and a local variable?
A field variable is a variable that is declared as a member of a class. A local variable is a variable that is declared local to a method.

Under what conditions is an object's finalize() method invoked by the garbage collector?
The garbage collector invokes an object's finalize() method when it detects that the object has become unreachable.

How are this() and super() used with constructors?
this() is used to invoke a constructor of the same class. super() is used to invoke a superclass constructor.

What is the relationship between a method's throws clause and the exceptions that can be thrown during the method's execution?
A method's throws clause must declare any checked exceptions that are not caught within the body of the method.

What is the difference between the JDK 1.02 event model and the event-delegation model introduced with JDK 1.1?
The JDK 1.02 event model uses an event inheritance or bubbling approach. In this model, components are required to handle their own events. If they do not handle a particular event, the event is inherited by (or bubbled up to) the component's container. The container then either handles the event or it is bubbled up to its container and so on, until the highest-level container has been tried.

In the event-delegation model, specific objects are designated as event handlers for GUI components. These objects implement event-listener interfaces. The event-delegation model is more efficient than the event-inheritance model because it eliminates the processing required to support the bubbling of unhandled events.

How is it possible for two String objects with identical values not to be equal under the == operator?
The == operator compares two objects to determine if they are the same object in memory. It is possible for two String objects to have the same value, but located indifferent areas of memory.

Why are the methods of the Math class static?
So they can be invoked as if they are a mathematical code library.

What Checkbox method allows you to tell if a Checkbox is checked?
getState()

What state is a thread in when it is executing?
An executing thread is in the running state.

What are the legal operands of the instanceof operator?
The left operand is an object reference or null value and the right operand is a class, interface, or array type.

How are the elements of a GridLayout organized?
The elements of a GridBad layout are of equal size and are laid out using the squares of a grid.

What an I/O filter?
An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another.

If an object is garbage collected, can it become reachable again?
Once an object is garbage collected, it ceases to exist.It can no longer become reachable again.

What is the Set interface?
The Set interface provides methods for accessing the elements of a finite mathematical set. Sets do not allow duplicate elements.

What classes of exceptions may be thrown by a throw statement?
A throw statement may throw any expression that may be assigned to the Throwable type.

What are E and PI?
E is the base of the natural logarithm and PI is mathematical value pi.

Are true and false keywords?
The values true and false are not keywords.

What is a void return type?
A void return type indicates that a method does not return a value.

What is the purpose of the enableEvents() method?
The enableEvents() method is used to enable an event for a particular object. Normally, an event is enabled when a listener is added to an object for a particular event. The enableEvents() method is used by objects that handle events by overriding their event-dispatch methods.

What is the difference between the File and RandomAccessFile classes?
The File class encapsulates the files and directories of the local file system. The RandomAccessFile class provides the methods needed to directly access data contained in any part of a file.

What happens when you add a double value to a String?
The result is a String object.

What is your platform's default character encoding?
If you are running Java on English Windows platforms, it is probably Cp1252. If you are running Java on English Solaris platforms, it is most likely 8859_1..

Which package is always imported by default?
The java.lang package is always imported by default.

What interface must an object implement before it can be written to a stream as an object?
An object must implement the Serializable or Externalizable interface before it can be written to a stream as an object.

How are this and super used?
this is used to refer to the current object instance. super is used to refer to the variables and methods of the superclass of the current object instance.

Explain the following keywords: super & final

1. super:

super keyword gives explicit access to the constructors, methods and variables of its superclass.

The super keyword works with inheritance. Inheritance gives implicit access to subclass to call its superclass members. But implicit access is blocked by method overriding or by declaring the same instance variables in subclass what superclass is having. Here, super keyword is used. That is **super keyword** gives explicit access to immediate superclass parts, even if access is blocked.

Usage of super with variables:
In the following program, the instance variable x is blocked by the subclass variable. Here **super** is used to print the superclass variable.

```
class A    {
       int x = 10 ;
}
public class B extends A    {
       in x = 20 ;
       public void display( )    {
              System.out.println( x ) ;                     // prints 20
              System.out.println( super.x ) ;          // prints 10
              System.out.println( x + super.x) ;      // prints 30
       }
       public static void main(String args[ ] )    {
              B b new B( ) ;
              b.display( ) ;
       }
}
```

Usage of super with methods:
In the following program, the display method is blocked by the subclass overridden method. Here **super** is used to call the superclass method.

```
class A    {
       public void display( )    {
              System.out.println( “ Hello1 “ ) ;
       }
}
public class B extends A    {
       public void display( )    {
              System.out.println( “ Hello2 “ ) ;
       }
       public void show( )    {
              display( ) ;                // Hello2 is printed
              super.display( ) ;             // Hello1 is printed
       }
```

```
        public static void main(String args[ ] )    {
                B b new B( ) ;
                b.show( );
        }
}
```

Usage of super with constructors - super( ) :

super keyword should be called in a different way with constructors, because **constructors are not inherited but accessed**. To access the superclass constructor from a subclass constructor we use super( ). JVM distinguishes which superclass consturctor is to be called by matching the parameters we pass with super( ). Following is the example :

```
class A    {
        A( )  {  System.out.println( “ Default A” ) ;
        A( int x )  {  System.out.println( x ) ;
}

class B extends A    {
        B( )  {  System.out.println( “ Default B” ) ;
        B( int x )  {
                super( 10 ) ;   // calls superclass A( int x ) constructor
                System.out.println( x ) ;
        }
        public static void main(String args[ ] )    {
                B b = new B( 5 ) ;
        }
}                          // output is 10 and 5
```

2. final :

final is a keyword in Java which generically means, **cannot be changed once created**. Any final keyword when declared with variables, methods and classes specifically means:

a final variable cannot be reassigned once initialized.
a final method cannot be overridden.
a final class cannot be subclassed.

Classes are usually declared final for either performance or security reasons. final methods work like inline code of C++.

final with variables:

final variables work like **const** of C-language that can't be altered in the whole program. That is, final variables once created can't be changed and they must be used as it is by all the program.

```
public static void main( String args[ ] )   {
        int x = 10;
        final int y = 20 ;
        x = 100 ;               // not an error as x is not final
        y = 200 ;               // error because y is final
}
```

final with methods:

Generally, a superclass method can be overriden by the subclass if it wants a different functionality. Or, it can use the same method if it wants the same functionality(output). If the superclass desires that the subclass shoud not override its method by the subclass, it declares the method as **final**. That is mehods declared final can not be overridden(else it is a compilation error).

```
class A    {
        final void display( )   {
                System.out.println( “ SNRao” ) ;
        }
}
class B extends A    {
        void display( )   {
                System.out.println( “ Sumathi ” ) ;
        }   // raises an error
}                // as display( ) is declared as final in the superclass
```

final with classes:

If you want the class not be subclassed by any other class, declare it **final**. Classes declared final can not be extended. That is, any class can use the methods of a final class by creating an object of final class and call the methods with its object.

```
final class A    {        }
class B extends A   {   }           // error because class A is declared final
```

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

What is a compilation unit?
A compilation unit is a Java source code file.

What interface is extended by AWT event listeners?
All AWT event listeners extend the java.util.EventListener interface.

What restrictions are placed on method overriding?
Overridden methods must have the same name, argument list, and return type. The overriding method may not limit the access of the method it overrides. The overriding method may not throw any exceptions that may not be thrown by the overridden method.

How can a dead thread be restarted?
A dead thread cannot be restarted.

What happens if an exception is not caught?
An uncaught exception results in the uncaughtException() method of the thread's ThreadGroup being invoked, which eventually results in the termination of the program in which it is thrown.

What is a layout manager?
A layout manager is an object that is used to organize components in a container.

Which arithmetic operations can result in the throwing of an ArithmeticException?
Integer / and % can result in the throwing of an ArithmeticException.

What are three ways in which a thread can enter the waiting state?
A thread can enter the waiting state by invoking its sleep() method, by blocking on I/O, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait() method. It can also enter the waiting state by invoking its (deprecated) suspend() method.

Can an abstract class be final?
An abstract class may not be declared as final.

What is the ResourceBundle class?
The ResourceBundle class is used to store locale-specific resources that can be loaded by a program to tailor the program's appearance to the particular locale in which it is being run.

What happens if a try-catch-finally statement does not have a catch clause to handle an exception that is thrown within the body of the try statement?
The exception propagates up to the next higher level try-catch statement (if any) or results in the program's termination.

What is numeric promotion?
Numeric promotion is the conversion of a smaller numeric type to a larger numeric type, so that integer and floating-point operations may take place. In numerical promotion, byte, char, and short values are converted to int values. The int values are also converted to long values, if necessary. The long and float values are converted to double values, as required.

What is the difference between a Scrollbar and a ScrollPane?

A Scrollbar is a Component, but not a Container. A ScrollPane is a Container. A ScrollPane handles its own events and performs its own scrolling.

What is the difference between a public and a non-public class?
A public class may be accessed outside of its package. A non-public class may not be accessed outside of its package.

To what value is a variable of the boolean type automatically initialized?
The default value of the boolean type is false.

Can try statements be nested?
Try statements may be tested.

What is the difference between the prefix and postfix forms of the ++ operator?
The prefix form performs the increment operation and returns the value ofthe increment operation. The postfix form returns the current value all of the expression and then performs the increment operation on that value.

What is the purpose of a statement block?
A statement block is used to organize a sequence of statements as a single statement group.

What is a Java package and how is it used?
A Java package is a naming context for classes and interfaces. A package is used to create a separate name space for groups of classes and interfaces. Packages are also used to organize related classes and interfaces into a single API unit and to control accessibility to these classes and interfaces.

What modifiers may be used with a top-level class?
A top-level class may be public, abstract, or final.

What are the Object and Class classes used for?
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.

What is an Object class ? Discuss some important methods of the class ?
Object is the root class for all Java classes. If a class does not extend any class, implicitly, the class extends class Object. Object class contains methods that dictate some common behavior for all Java classes. Object class is defined in java.lang package that is implicitly imported.

Some important methods of Object class:
finalize( ) : This method is called implicitly by the garbage collector before an object is garbage collected. In Java, an object is liable for garbage collection, if no references exist for it in the remaining part of the program. It's access modifier is protected and return void. Following is the method signature specified in the Object class:

```
protected void finalize( ) throws Throwable
```

toString( ) : This method converts any object into a string form. It's return type is a String.

Following is the method signature specified in the Object class:

```
public String toString( )
```

Example:

```
StringBuffer sb = new StringBuffer( );
String str = sb.toString( ) ;
```

wait( ) : This method is used in thread synchronization. Synchronization helps integrity of data that is shared in between concurrently running threads. Locking of a shared resource is necessary to have consistent(or durable) data. In monitors, when a thread tries to access an object which already being accessed, wait( ) method is called and thread waits until notify( ) is called. The following is the method signature as defined in Object class:

```
public final void wait( ) throws InterruptedException
```

This method cannot be overridden in its subclasses as it final. wait( ) method is overloaded.

notify( ) and notifyAll( ) : In thread synchronization, notify( ) or notifyAll( ) is called on the waiting thread(s) to get access to the synchronized code. The following is the method signature as defined in Object class:

public final void notify( )
public final void notifyAll( )

notify( ) is called on a single waiting thread, notifying the thread that the condition of the synchronized code to be accessed is changed. notifyAll( ) is called when threads are many waiting in the queue. The above methods cannot be overridden in the subclasses as they are final.

How does a try statement determine which catch clause should be used to handle an exception?
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 exception is executed. The remaining catch clauses are ignored.

Can an unreachable object become reachable again?
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.

When is an object subject to garbage collection?
An object is subject to garbage collection when it becomes unreachable to the program in which it is used.

What method must be implemented by all threads?
All tasks must implement the run() method, whether they are a subclass ofThread or implement the Runnable interface.

What methods are used to get and set the text label displayed by a Button object?
getLabel() and setLabel()

Which Component subclass is used for drawing and painting?
Canvas

What are synchronized methods and synchronized statements?
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.

What are the two basic ways in which classes that can be run as threads may be defined?
A thread class may be declared as a subclass of Thread, or it may implement the Runnable interface.

What are the problems faced by Java programmers who don't use layout managers?
Without layout managers, Java programmers are faced with determining how their GUI will be displayed across multiple windowing systems and finding a common sizingand positioning that will work within the constraints imposed by each windowing system.

What is the difference between an if statement and a switch statement?
The if statement is used to select among two alternatives. It uses a boolean expression to decide which alternative should be executed. The switch statement is used to select among multiple alternatives. It uses an int expression to determine which alternative should be executed.

What happens when you add a double value to a String?
The result is a String object.

What is the List interface?
The List interface provides support for ordered collections of objects.

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.

Why would you use a synchronized block vs. synchronized method?
Synchronized blocks place locks for shorter periods than synchronized methods.

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).

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.

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 supeclass type, the casting is performed automatically.

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.

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.

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.

Can you call one constructor from another if a class has multiple constructors
Yes. Use this() syntax.

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.

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

What's the difference between J2SDK 1.5 and J2SDK 5.0?
There's no difference, Sun Microsystems just re-branded this version.

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.

Does it matter in what order catch statements for FileNotFoundException and IOExceptipon are written?
Yes, it does. The FileNoFoundException is inherited from the IOException. Exception's subclasses have to be caught first.

Can an inner class declared inside of a method access local variables of this method?
It's possible if these variables are final.

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.

What's the main difference between a Vector and an ArrayList
Java Vector class is internally synchronized and ArrayList is not.

When should the method invokeLater()be used?
This method is used to ensure that Swing components are updated through the event-dispatching thread.

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.

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

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.

What comes to mind when you hear about a young generation in Java?
Garbage collection.

What comes to mind when someone mentions a shallow copy in Java?
Object cloning.

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

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

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().

How can you minimize the need of garbage collection and make the memory use more effective?
Use object pooling and weak object references.

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.

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.

What's the difference between constructors and normal 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 and it can return a value or can be void.

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.

Describe the three OOO principles ?
Encapsulation - It is the way the code and data are confined and are in isolation from the outside environment of the system.
EXAMPLE: In a car the engine can be thought about as an encapsulated which is controlled by the starter key and the gear. The operation of the engine does not affect the functioning of other parts of car like the headlight and wiper. In JAVA basis of encapsulation is the CLASS.

Inheritance - It is the process by which one object acquires the properties of another object.

Polymorphism - It is a feature that allows one interface to be used for a general class of actions. The specific action is determined by the exact nature of the situation.
EXAMPLE: In a college cafeteria if an old music is put the students would not like it , but if the latest number of Brittney Spears is put then they would love it. If analyzed then we can see that even though the process of listening is the same i.e. for the old song the students listen it from there ears and even for the latest Brittney Spears number they listen it from there ear. But there is a difference by which the students react. This difference can be explained by the concept polymorphism.

How many types of literals are there in JAVA ?
There are four types of literals they are Integer literals, Floating point literals, Boolean literals and character literals.

A note on compiling & executing a JAVA pgm ?
(i) The name of the source file is called in terms of .java
(ii) A source file is called a compilation unit. This has one or more class definitions.
(iii) The name of the class should be same as that of the file.
(iv) Once compiled the .java file creates a .class file. This is done by the compiler javac
(v) This class file contains the byte code version of the program.

A note on PUBLIC, PRIVATE, STATIC, VOID & MAIN ?
(i) All Java applications begin execution by calling main ()
(ii) When a class member is defined as public. Then that member may be accessed by code outside the class in which it is declared.
(iii) The opposite of public is private which prevents a member from being used by code defined outside of its class.
(iv) The keyword static allows main() to be called without having to instantiate a particular instance of the class. This is mandatory because main () is called by the Java interpreter before any objects are made.
(v) CASE SENSITIVE: Main () is different from main(). It is important to know that that Main() would be compiled. But the Java interpreter would report an error if it would not find main().

What is meant by Garbage collection ?
The technique that automatically destroys the dynamically created objects is called garbage collection. When no reference to an object exists, that object is assumed to be no longer needed, and memory occupied by that object can be reclaimed.

What are the access modifiers ?
There are three types of access modifiers.
(i) Private - Makes a method or a variable accessible only from within its own class.
(ii) Protected - Makes a method or a variable accessible only to classes in the same package or subclasses of the class.
(iii) Public - Makes a class , method or variable accessible from any other class.

A note on keywords for Error handling ?
(i) Catch - Declares the block of code used to handle an exception.
(ii) Finally - Block of code , usually following a typecatch statement, which is executed no matter what program flow occurs when dealing with an exception.
(iii) Throw - Used to pass an exception up to the method that calls this method.
(iv) Throws - Indicates the method will pass an exception to the method that called it.
(v) Try - Block of code that will be tried but which may cause an exception.
(vi) Assert - Evaluates a conditional _expression to verify the programmer's assumption.

How many ways can you represent integer numbers in JAVA ?
There are three ways , you can represent integer numbers in JAVA. They are decimal (base 10) , octal (base 8) , and hexadecimal (base 16).

A note on defining floating point literal ?
A floating point literal is defined as float g = 3576.2115F.

What is meant by "instanceof" comparison ?
It is used for object reference variables only. You can use it to check whether an object is of a particular type.

when is a method said to be overloaded ?
Two or more methods are defined within the same class that share the same name and their parameter declarations are different then the methods are said to be overloaded.

What is meant by Recursion ?
It is the process of defining something in terms of itself. In terms of JAVA it is the attribute that allows a method to call itself. The following example of calculating a factorial gives an example of recursion.

```
class Factorial {
        int fact (int n) {
                int result;
                if (n= 1) return 1;
                result = fact(n -1) * n;
                return result;
        }
}
class Recursion {
        Public static void main (string args[ ]) {
                Factorial f = new Factorial ();
                system.out.println ("Factorial of 10 is " + f.fact(10));
        }
}
```

Difference between a process and a thread ?
"Threads are like tiny ropes. A Process would denote a rope made out from these threads." Fun apart, a process can contain multiple threads. Also a process gets its own memory address space while a thread doesn't.

What are checked and unchecked exceptions ?
Checked exceptions are the ones which you expect beforehand to be raised when an exceptional condition occurs and so write your code in a try-catch block to handle that sufficiently. For example: InsuffucientBalanceException which might be raised when money is being withdrawn from a bank account and the account has insufficient balance. Checked exceptions are sub classes of Exception.

All the subclasses of Exception class except RuntimeException are called **checked exceptions**. For these exceptions, programmer should provide a try – catch block whether really exception raises or not, else program

does not compile. Generally, these exceptions are written in the method signature itself with **throws** clause. Following are the examples from Java API:

```
public FileInputStream( String filename ) throws FileNotFoundException
public void sleep( int milliseconds ) throws InterruptedException
```

In the above statements, FileNotFoundException and InterruptedException are checked exceptions. That is, when FileInputStream(String filename) constructor is used by the programmer, he should write a try – catch mechanism for FileNotFoundException whether the file passed as parameter really exists in the hard disk or not.

Unchecked exceptions are the ones which cannot be handled in the code. These are rather unexpected exceptions like NullPointerException, OutOfMemoryError, DivideByZeroException, typically, programming errors. Unchecked exceptions are subclasses of RunTimeExceptions.

All the subclasses of **RuntimeException** are called **unchecked exceptions**. Some examples are ArithmeticException and ArrayIndexOutOfBoundsException. These exceptions are called unchecked for the reason, even if the programmer does not provide any try – catch mechanism, the **program compiles**. System checks them at runtime and if an exception occurs program terminates. If a try – catch block is provided for these exceptions, program execution continues further when exception really occurs.

Is synchronised a modifier?identifier? what is it ?
It's a modifier. 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.

What is singleton class? Where is it used ?
Singleton is a design pattern meant to provide one and only one instance of an object. Other objects can get a reference to this instance through a static method (class constructor is kept private). Why do we need one? Sometimes it is necessary, and often sufficient, to create a single instance of a given class. This has advantages in memory management, and for Java, in garbage collection. Moreover, restricting the number of instances may be necessary or desirable for technological or business reasons--for example, we may only want a single instance of a pool of database connections.

How to avoid deadlock?
Don't go to sleep holding a lock Coordinate thread access to synchronized methods using wait and notify.

Garbage collection thread belongs to which priority...Min?normal?max ?
The Java garbage collection is implemented as a low priority thread

What is meant by time slicing?
Its a task scheduling method. With time slicing, or "Round-Robin Systems", several processes are executed sequentially to completion. Each executable task is assigned a fixed-time quantum called a time slice in which to execute.

What is a compilation unit?
The smallest unit of source code that can be compiled, i.e. a .java file.

Explain the Inheritance principle?
Inheritance is the process by which one object acquires the properties of another object.

Explain the Polymorphism principle?
The meaning of Polymorphism is something like one name many forms. Polymorphism enables one entity to be used as general category for different types of actions. The specific action is determined by the exact nature of the situation. The concept of polymorphism can be explained as "one interface, multiple methods".

Explain the different forms of Polymorphism?

From a practical programming viewpoint, polymorphism exists in three distinct forms in Java:

- Method overloading
- Method overriding through inheritance
- Method overriding through the Java interface

Describe the wrapper classes in Java?

Wrapper class is wrapper around a primitive data type. An instance of a wrapper class contains, or wraps, a primitive value of the corresponding type. Following table lists the primitive types and the corresponding wrapper classes:

- Primitive Wrapper
- boolean java.lang.Boolean
- byte java.lang.Byte
- char java.lang.Character
- double java.lang.Double
- float java.lang.Float
- int java.lang.Integer
- long java.lang.Long
- short java.lang.Short
- void java.lang.Void

What is the class variables ?

When we create a number of objects of the same class, then each object will share a common copy of variables. That means that there is only one copy per class, no matter how many objects are created from it. Class variables or static variables are declared with the static keyword in a class, but mind it that it should be declared outside a class. These variables are stored in static memory. Class variables are mostly used for constants, variable that never change its initial value. Static variables are always called by the class name. This variable is created when the program starts i.e. it is created before the instance is created of class by using new operator and gets destroyed when the programs stops. The scope of the class variable is same a instance variable. The class variable can be defined anywhere at class level with the keyword static. It initial value is same as instance variable. When the class variable is defined as int then it's initial value is by default zero, when declared Boolean its default value is false and null for object references. Class variables are associated with the class, rather than with any object.

What is the difference between the instanceof and getclass, these two are same or not ?

instanceof is a operator, not a function while getClass is a method of java.lang.Object class. Consider a condition where we use if(o.getClass().getName().equals("java.lang.Math")){ } This method only checks if the classname we have passed is equal to java.lang.Math. The class java.lang.Math is loaded by the bootstrap ClassLoader. This class is an abstract class.This class loader is responsible for loading classes. Every Class object contains a reference to the ClassLoader that defines. getClass() method returns the runtime class of an object. It fetches the java instance of the given fully qualified type name. The code we have written is not necessary, because we should not compare getClass.getName(). The reason behind it is that if the two different class loaders load the same class but for the JVM, it will consider both classes as different classes so, we can't compare their names. It can only gives the implementing class but can't compare a interface, but instanceof operator can. The instanceof operator compares an object to a specified type. We can use it to test if an object is an instance of a class, an instance of a subclass, or an instance of a class that implements a particular interface. We should try to use instanceof operator in place of getClass() method. Remember instanceof opeator and getClass are not same. Try this example, it will help you to better understand the difference between the two.

```
Interface one{}

Class Two implements one { }

Class Three implements one {}

public class Test {
        public static void main(String args[]) {
                one test1 = new Two();
                one test2 = new Three();
                System.out.println(test1 instanceof one); //true
```

```
                    System.out.println(test2 instanceof one); //true
                    System.out.println(Test.getClass().equals(test2.ge tClass())); //false
            }
    }
```

How can you reference static variables?
Via reference to any instance of the class
Code:

```
    Computer comp = new Computer ();
    comp.harddisk where hardisk is a static variable
    comp.compute() where compute is a method
    Via the class name
    Code:
    Computer.harddisk
    Computer.compute()
```

Can static method use non static features of there class ?
No they are not allowed to use non static features of the class, they can only call static methods and can use static data

What do you mean by " Can't make a static reference to a nonstatic variable" ?
We cannot call an instance variable without an object(instance). But we can call a static variable(also called class variable) without the help of a object.

" Can't make a static reference to a nonstatic variable" is a common compilation error that is flashed when we generally try to call an instance variable without an object. Here, main is static and variable is nonstatic. Declare the variable static, and this error is not flashed.

I. The following code works without a error as variable and method are nonstatic.

```
        int x = 10 ;                    // nonstatic variable
        void display( )   {             // nonstatic method
                System.out.println( x ) ;
        }
```

2. The following code does not work as variable is nonstatic and method is static.

```
        int x = 10 ;                    // nonstatic variable
        static void display( )   {              // static method
                System.out.println( x ) ;
        }
```

What is static initializer code ?
A class can have a block of initializer code that is simply surrounded by curly braces and labeled as static e.g.
Code:

```
    public class Demo{
    static int =10;
    static{
    System.out.println(ï¿½Hello worldï¿½);
    }
    }
```

And this code is executed exactly once at the time of class load

Where is native modifier used ?
It can refer only to methods and it indicates that the body of the method is to be found else where and it is usually written in non java language

What are transient variables ?
A transient variable is not stored as part of objects persistent state and they cannot be final or static

What is synchronized modifier used for ?
It is used to control access of critical code in multithreaded programs

What are volatile variables ?
It indicates that these variables can be modified asynchronously

What are the rules for primitive arithmetic promotion conversion ?
For Unary operators :
If operant is byte, short or a char it is converted to an int. If it is any other type it is not converted
For binary operands :
If one of the operands is double, the other operand is converted to double
Else If one of the operands is float, the other operand is converted to float
Else If one of the operands is long, the other operand is converted to long
Else both the operands are converted to int

What are the rules for casting primitive types ?
You can cast any non Boolean type to any other non boolean type. You cannot cast a boolean to any other type; you cannot cast any other type to a boolean

What are the rules for object reference assignment and method call conversion ?
An interface type can only be converted to an interface type or to object. If the new type is an interface, it must be a superinterface of the old type.
A class type can be converted to a class type or to an interface type. If converting to a class type the new type should be superclass of the old type. If converting to an interface type new type the old class must implement the interface.
An array maybe converted to class object, to the interface cloneable, or to an array. Only an array of object references types may be converted to an array, and the old element type must be convertible to the new element

What are the rules for Object reference casting?
Casting from Old types to Newtypes
Compile time rules :
- When both Oldtypes and Newtypes are classes, one should be subclass of the other
- When both Oldtype ad Newtype are arrays, both arrays must contain reference types (not primitive), and it must be legal to cast an element of Oldtype to an element of Newtype
- You can always cast between an interface and a non-final object
Runtime rules :
- If Newtype is a class. The class of the expression being converted must be Newtype or must inherit from Newtype
- If NewType is an interface, the class of the expression being converted must implement Newtype

When do you use continue and when do you use break statements ?
When continue statement is applied it prematurely completes the iteration of a loop. When break statement is applied it causes the entire loop to be abandoned.

What is the base class from which all exceptions are subclasses ?
All exceptions are subclasses of a class called java.lang.Throwable

How do you intercept and thereby control exceptions ?
We can do this by using try/catch/finally blocks
You place the normal processing code in try block
You put the code to deal with exceptions that might arise in try block in catch block
Code that must be executed no matter what happens must be place in finally block

When do we say an exception is handled ?
When an exception is thrown in a try block and is caught by a matching catch block, the exception is considered to have been handled

When do we say an exception is not handled ?

There is no catch block that names either the class of exception that has been thrown or a class of exception that is a parent class of the one that has been thrown, then the exception is considered to be unhandled, in such condition the execution leaves the method directly as if no try has been executed

In what sequence does the finally block gets executed ?
If you put finally after a try block without a matching catch block then it will be executed after the try block
If it is placed after the catch block and there is no exception then also it will be executed after the try block
If there is an exception and it is handled by the catch block then it will be executed after the catch block

What can prevent the execution of the code in finally block ?
- The death of thread
- Use of system.exit()
- Turning off the power to CPU
- An exception arising in the finally block itself

What are the rules for catching multiple exceptions
- A more specific catch block must precede a more general one in the source, else it gives compilation error
- Only one catch block, that is first applicable one, will be executed

What does throws statement declaration in a method indicate ?
This indicates that the method throws some exception and the caller method should take care of handling it

What are checked exception?
Checked exceptions are exceptions that arise in a correct program, typically due to user mistakes like entering wrong data or I/O problems

What are runtime exceptions?
Runtime exceptions are due to programming bugs like out of bond arrays or null pointer exceptions.

What is difference between Exception and errors?
Errors are usually compile time and exceptions can be runtime or checked

How will you handle the checked exceptions?
You can provide a try/catch block to handle it. OR
Make sure method declaration includes a throws clause that informs the calling method an exception might be thrown from this particular method
When you extend a class and override a method, can this new method throw exceptions other than those that were declared by the original method
No it cannot throw, except for the subclasses of those exceptions

Is it legal for the extending class which overrides a method which throws an exception, not o throw in the overridden class ?
Yes it is perfectly legal

Explain modifier final?
Final can be applied to classes, methods and variables and the features cannot be changed. Final class cannot be subclassed, methods cannot be overridden

Why is the main method static?
So that it can be invoked without creating an instance of that class

What is the difference between class variable, member variable and automatic(local) variable ?
- class variable is a static variable and does not belong to instance of class but rather shared across all the instances
- member variable belongs to a particular instance of class and can be called from any method of the class
- automatic or local variable is created on entry to a method and has only method scope

When are static and non static variables of the class initialized?
The static variables are initialized when the class is loaded Non static variables are initialized just before the constructor is called

When are automatic variable initialized?
Automatic variable have to be initialized explicitly

What is a modulo operator %?
This operator gives the value which is related to the remainder of a divisione.g x=7%4 gives remainder 3 as an answer

How is an argument passed in java, by copy or by reference What is a modulo operator %?
This operator gives the value which is related to the remainder of a divisione.g x=7%4 gives remainder 3 as an answer

What is garbage collection?
The runtime system keeps track of the memory that is allocated and is able to determine whether that memory is still useable. This work is usually done in background by a low-priority thread that is referred to as garbage collector. When the gc finds memory that is no longer accessible from any live thread it takes steps to release it back to the heap for reuse

Does System.gc and Runtime.gc() guarantee garbage collection ?
No

What are different types of operators in Java ?
- Uniary ++, –, +, -, |, ~, ()
- Arithmetic *, /, %,+, -
-Shift <<, >>, >>>
- Comparison =, instanceof, = =,!=Bitwise &, ^, |Short Circuit &&, ||Ternary ?:Assignment =

How does bitwise (~) operator work ?
It converts all the 1 bits in a binary value to 0s and all the 0 bits to 1s, e.g 11110000 coverts to 00001111

Can shift operators be applied to float types ?
No, shift operators can be applied only to integer or long types

What happens to the bits that fall off after shifting ?
They are discarded

What values of the bits are shifted in after the shift ?
In case of signed left shift >> the new bits are set to zero. But in case of signed right shift it takes the value of most significant bit before the shift, that is if the most significant bit before shift is 0 it will introduce 0, else if it is 1, it will introduce 1

Can protected or friendly features be accessed from different packages ?
No when features are friendly or protected they can be accessed from all the classes in that package but not from classes in another package

How can you access protected features from another package ?
You can access protected features from other classes by subclassing the that class in another package, but this cannot be done for friendly features

What are the rules for overriding ?
Private method can be overridden by private, friendly, protected or public methods. Friendly method can be overridden by friendly, protected or public methods. Protected method can be overridden by protected or public methods. Public method can be overridden by public method

Can you change the reference of the final object ?
No the reference cannot be change, but the data in that object can be changed

Can abstract modifier be applied to a variable ?
No it is applied only to class and methods

Can abstract class be instantiated ?
No abstract class cannot be instantiated i.e you cannot create a new object of this class

When does the compiler insist that the class must be abstract ?
If one or more methods of the class are abstract. If class inherits one or more abstract methods from the parent abstract class and no implementation is provided for that method If class implements an interface and provides no implementation for those methods

How is abstract class different from final class ?
Abstract class must be subclassed and final class cannot be subclassed

Where can static modifiers be used ?
They can be applied to variables, methods and even a block of code, static methods and variables are not associated with any instance of class

When are the static variables loaded into the memory ?
During the class load time

When are the non static variables loaded into the memory ?
They are loaded just before the constructor is called

Is string a wrapper class ?
String is a class, but not a wrapper class. Wrapper classes like (Integer) exist for each primitive type. They can be used to convert a primitive data value into an object, and vice-versa.

how can you retrieve warning in jdbc ?
Write my JDBC code in a try-catch block and catch the SQLExcpetions

Is there any tool in java that can create reports ?
Yes there are third party tools available.

Why java does not have multiple inheritance ?
The Java design team strove to make Java: Simple, object oriented, and familiar Robust and secure Architecture neutral and portable High performance Interpreted, threaded, and dynamic The reasons for omitting multiple inheritance from the Java language mostly stem from the "simple, object oriented, and familiar" goal. As a simple language, Java's creators wanted a language that most developers could grasp without extensive training. To that end, they worked to make the language as similar to C++ as possible (familiar) without carrying over C++'s unnecessary complexity (simple). In the designers' opinion, multiple inheritance causes more problems and confusion than it solves. So they cut multiple inheritance from the language (just as they cut operator overloading). The designers' extensive C++ experience taught them that multiple inheritance just wasn't worth the headache.

Why java is not a 100% oops ?
In a 100% Object oriented language, everything is an Object. That is not the case with Java. Java uses primitive types such as int, char, double. Java primitive types ARE NOT OBJECTS. All the rest are objects. Keep in mind that int, char and other primitive types CAN NOT BE STORED in collections (e.g. Vector, Hashtable) to store these types in a collections, you need to Use wrapper classes (e.g. Integer, Double).

What is a resource bundle ?
In its simplest form, a resource bundle is represented by a text file containing keys and a text value for each key.

What is meant by resource leak?

Resource leakage generally refers to memory leakage but can refer to any type of system resource that isn't managed correctly. Memory leakage refers to memory that is no longer used but is not freed and therefore is not available for the system to reuse. Java is supposed to remove most of the memory leakage issues of other languages but it is still possible to create situations that waste memory such as java.util.Map that you never remove information from. Resource leakage in a more general sence can refer to any system or external application finite resource. The java.awt.Graphics context is supposed to be like this because in java it is a wrapper on a system resource. If you don't close it, it can't be cleaned up by the system. Database resources are another example. Poor Databse code can prevent connections from being recycled by the DBMS and represent a drag on the database.

What do you understand by Synchronization ?

Synchronization is a process of controlling the access of shared resources by the multiple threads in such a manner that only one thread can access one resource at a time. In non synchronized multithreaded application, it is possible for one thread to modify a shared object while another thread is in the process of using or updating the object's value. Synchronization prevents such type of data corruption.

E.g. Synchronizing a function:

```
        public synchronized void Method1 () {
        // Appropriate method-related code.
        }

E.g. Synchronizing a block of code inside a function:
        public myFunction (){
                synchronized (this) {
                // Synchronized code here.
                }
        }
```

What is Collection API ?

The Collection API is a set of classes and interfaces that support operation on collections of objects. These classes and interfaces are more flexible, more powerful, and more regular than the vectors, arrays, and hashtables if effectively replaces.

Example of classes: HashSet, HashMap, ArrayList, LinkedList, TreeSet and TreeMap.

Example of interfaces: Collection, Set, List and Map.

- A Class may implement several interfaces. But in case of abstract class, a class may extend only one abstract class.
- Interfaces are slow as it requires extra indirection to find corresponding method in the actual class. Abstract classes are fast.

Similarities:

- Neither Abstract classes or Interface can be instantiated.

How to define an Abstract class ?

A class containing abstract method is called Abstract class. An Abstract class can't be instantiated.

```
        Example of Abstract class:
        abstract class testAbstractClass {
        protected String myString;
        public String getMyString() {
        return myString;
        }

        public abstract string anyAbstractFunction();
        }
```

How to define an Interface ?

In Java Interface defines the methods but does not implement them. Interface can include constants. A class that implements the interfaces is bound to implement all the methods defined in Interface.

```
        Example of Interface:
```

```
public interface sampleInterface {
public void functionOne();
public long CONSTANT_ONE = 1000;
}
```

Explain the user defined Exceptions ?

User defined Exceptions are the separate Exception classes defined by the user for specific purposed. An user defined can created by simply sub-classing it to the Exception class. This allows custom exceptions to be generated (using throw) and caught in the same way as normal exceptions.

Example:

```
class myCustomException extends Exception {
// The class simply has to exist to be an exception
}
```

Explain garbage collection ?

Garbage collection is one of the most important feature of Java. Garbage collection is also called automatic memory management as JVM automatically removes the unused variables/objects (value is null) from the memory. User program cann't directly free the object from memory, instead it is the job of the garbage collector to automatically free the objects that are no longer referenced by a program. Every class inherits finalize() method from java.lang.Object, the finalize() method is called by garbage collector when it determines no more references to the object exists. In Java, it is good idea to explicitly assign null into a variable when no more in use. I Java on calling System.gc() and Runtime.gc(), JVM tries to recycle the unused objects, but there is no guarantee when all the objects will garbage collected.

How you can force the garbage collection?

Garbage collection automatic process and can't be forced.