1. **What is JAVA ?**

Java is a programming language used to develop different applications like standalone application,Web application, Enterprise application and Mobile application.it also has some important features like

Simple , object oriented ,robust, platform independent, distributed , multithreading etc.,

1. **Features of JAVA ?**

Java is a programming language which supports different features simple, object oriented robust, platform independent ,portable, secured , multithreaded and distributed etc.,

* **Simple** : Simple- Java was designed to be easy for the professional programmer to learn and use it effectively. Since Java inherits C/C++ syntax and object oriented (OO) concepts of C++, it is easy to learn for most programmers.
* **Object** **oriented** : java follows different object oriented concepts like inheritance abstraction, encapsulation and polymorphism .
* **Robust** : Robust simply means strong. Java uses strong memory management. There are lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points makes java robust.
* **Plaform** **Independent** :The byte code generated by a valid java compiler can be carried to any platform.
* **Portable** :Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA).
* **Secured** : Java is secured because,
  + - * + No explicit pointer.
        + Java Programs run inside virtual machine sandbox.
    - **Multithreaded :**  We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area.

1. **What are the Different types of applications you can build with JAVA ?**

There are mainly 4 type of applications that can be created using java programming:

**1) Standalone Application:**It is also known as desktop application or window-based application. An application that we need to install on every machine such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.

**2) Web Application:**

An application that runs on the server side and creates dynamic page, is called web application. Currently, servlet, jsp, struts, jsf etc. technologies are used for creating web applications in java.

**3) Enterprise Application:**

An application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

**4) Mobile Application:**

An application that is created for mobile devices. Currently Android and Java ME are used for creating mobile applications.

**4. Difference between JDK, JRE and JVM ?**

JVM (Java Virtual Machine) : It is an abstract machine which loads, verifies and executes the code, provides the run time environment for the byte code to be executed. It directly comes in contact with OS but the code within the JVM never interacts with it which makes it more secured.

JRE (Java Run Time Environment): It is the implementation of the JVM. It has got all the libraries + all the files used by JVM at run time. It provides the run time environment.

JDK (Java Development Kit): It provides the utilities for the compilation of the code and includes development tools. It consists of the JRE.

1. **What is the Java memory Model?**

* Class Area: It consists of the static variables.
* Heap: The objects created at the run time are stored in heap.
* Stack: It holds the primitive variables, partial results and local variables.
* Program Counter: It consists of the address of the current instruction being executed.
* Native Methods: it consists of the methods of other programming languages like C.

1. **What are the types of variables in JAVA?**

* Local Variables: These variables are declared within a method and their scope is also within the method. They don’t have any default values.
* Static Variables: The variable declared as static are known as static variables. It cannot be local.
* Instance Variables: The variables declared outside a method but inside a class are known as instance variables. They do have default values like int (0), string (null) etc.

1. **Different types of operators ?**

There are many types of operators in java which are given below:

* + - * + Unary Operator,
        + Arithmetic Operator,
        + shift Operator,
        + Relational Operator,
        + Bitwise Operator,
        + Logical Operator,
        + Ternary Operator and
        + Assignment Operator.

1. **Explain the design of JAVA class ?**

The design of java class starts with valid package name ,import statements ,class name , variable declaration ,method signature , comments

1. **What is Class and Object ?**

Class : class is a template which represents state and behavior . here the state incudes properties and behavior includes methods .

Object: object is an instance of class.

1. **What are different OOPS concepts?**

* Abstraction
* Polymorphism
* Encapsulation
* Inheritance

1. **What is Abstraction?**

Abstraction: Abstraction is the **concept** of hiding the implementation and showing only the essential features of the object. Abstraction mainly comes into picture for future flexibility when the developer is not sure of the implementation.

A class that is declared as abstract is known as **abstract class**. It needs to be extended and its method implemented. It cannot be instantiated. A method that is declared as abstract and does not have implementation is known as abstract method.

1. **What is Encapsulation?**

Encapsulation is the **concept** of restricting the access of data members from one class to another by giving the private keyword and providing access through the getters and setters.

1. **What is the difference between abstraction and encapsulation?**

**Encapsulation**is a **concept** by which you restrict the access to some of the object's components, as well as binding the data and methods operating on the data.

**Abstraction** is the **concept** to define an object that can represent abstract entities which can work, change state and communicate with other entities.

**Abstraction** is implemented in Java using interface and abstract class while

**Encapsulation** is implemented using private and protected access modifier

1. **What is Inheritance?**

Inheritance is the **concept** of acquisition of properties and behavior from one class to another. It is mainly used for code reusability. It is an IS-A relationship. The extends keyword is used to extend once class to another. A class that is inherited is called a super class. The new class is called a subclass.

1. **What is Polymorphism?**

Polymorphism is the **concept** of an object’s ability to take on many forms. There are two types of polymorphism in java: compile time polymorphism and runtime polymorphism.

1. **What is method overloading and overriding?**

Method overloading is the **concept** of a class having multiple methods of same name but vary with type and number of arguments. It enhances the readability. It is performed within a class.

Method overriding is the **concept** of implementing a method in the sub class which is already provided in the super class. It should compulsorily be in an IS-A relation. The subclass method should have the same method name and the parameters.

1. **What is static in JAVA?**

Static is a **keyword** which specifies that the variables, keywords will have common properties to all the instances of the class, rather than depending on the object. It can be applied to a variable, method, block or a nested class. It is mainly used for memory management.

1. **What is final ?**

FINAL is keyword which can be applied towards classes, variables and methods

When we apply towards classes we can’t inherit the classes , when we apply towards variables the value of variable cannot be reassigned or modified, when we apply towards methods we can’t override the methods

1. **What is final, finally, finalize ?**

**FINAL** is keyword which can be applied towards classes, variables and methods When we apply towards classes we can’t inherit the classes , when we apply towards variables the value of variable cannot be reassigned or modified, when we apply towards methods we can’t override the methods

**Finally**  is a block where we place important statements of code which needs to be executed irrespective of occurrence of exception. Example closing database connections, file reader, buffered reader connection .

**Finalize**  is method which is used to perform clean up processing before object is garbage collection .

1. **What is static and dynamic binding ?**

The concept of establishing connection between method call and method body is called binding

There are two types of binding

**Static binding :** when the object type is determined at compile time then it is called static binding.

**Dynamic binding :** when the object type is determined at runtime it is called dynamic binding

1. **What is Heap space in Java?**

When a Java program started Java Virtual Machine gets some memory from Operating System. Java Virtual Machine or JVM uses this memory for all its need and part of this memory is call java heap memory. Heap in Java generally located at bottom of address space and move upwards. whenever we create object using new operator or by any another means object is allocated memory from Heap and When object dies or garbage collected ,memory goes back to Heap space in Java  
Read more: <http://javarevisited.blogspot.com/2011/05/java-heap-space-memory-size-jvm.html#ixzz4bbDz5J2D>

1. **Abstract class vs interface ?**

**Abstract class: -**

* Can have abstract and non-abstract methods.
* Doesn’t support multiple inheritance.
* Can have final, non-final, static, non-static variables.
* Can have static method, main method and constructor.
* Can provide implementation to interface.
* Abstract keyword is used to declare abstract class.

**Interface: -**

* Can have only abstract methods.
* Support multiple inheritance.
* Can have only static and final variable.
* Cannot have static methods, main method and constructor.
* Cannot provide implementation of abstract class.
* Interface keyword is used to declare interface.

1. **Why do you create abstract classes in application development?**

When the developer is not sure about the implementation and for the future flexibility of code.

1. **Why do you create interfaces in application development?**

When multiple classes have similar functionality but vary in implementation of that particular functionality then interface comes into picture.

1. **What are different types of access modifiers?**

**Public**: Accessible everywhere.

**Private**: Accessible within the class.

**Protected**: Accessible within the package and outside the package only through inheritance.

**Default**: Accessible only within the package.

1. **What is call by value?**

Whenever we call a method by passing primitive data-types the value itself is copied into method signature. So, changes made to a variable within the variable has no effect on the variable that is passed .When objects are passed , no copy is made.

1. **What is string pool?**

String’s are stored in special memory area inside the heap memory. String is possible only because of immutability concept of string. When-ever String iteral is created JVM looks for String with same value if found it just returns the reference else it creates a new String in the pool and returns the reference.

1. **Why string is immutable?**

As majority of data is represented in the form of String. There is high chances of String pointing towards same reference. If a string changes, it will affect the other string which is being referenced to the same pointer and there are chances that it might become unreferenced which gets deleted by the automatic garbage collector. This is the reason why strings are made immutable.

1. **How do you make a class immutable?**

By declaring a class final.

By declaring the properties of the class as final.

By not providing the setters.

1. **String Buffer vs String Builder?**

* **StringBuffer** and StringBuilder are classes in java. Which provides some methods to perform Operation to manipulate Strings.
* **StringBuffer**:- StringBuffer is synchronized i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously. So, it is performances is low.
* **StringBuilder**:- StringBuilder is non-synchronized i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously. So, it is performances is high.

1. **== vs Equals?**

Both == and Equals are methods in java. In which == method check for reference of the String. Equals method checks for content of the String.

1. **What is exception handling and how do you achieve it?**

Firstly, exception is an abnormal condition which intercepts the normal flow of program and application may terminate abnormally. In order to avoid it we use exception handling mechanism and it is achieved by by some keywords like try,catch,throw and throws. In **try** block we if some part of code rises exception it will be handled by catch block. If there is no exception in try block catch block will not be executed. **Throws** we declare an exception in method signature generally checked exception goes here. So, it is caller of the method responsibility to handle it by using try ,catch keywords. **Throw** we explicity throw the exception. Generally those are checked exceptions.

1. **How do you make sure a code must be executed even if exception happens?**

By placing the code in finally block because it will get executed even if exception is handled or not. Generally, closing the connections types of code goes here.

1. **What code you normally write in finally block?**
2. **What are checked and unchecked exceptions?**

**Checked Exceptions:-** Exceptions which occur at compile time is known as checked exceptions. Generally these exceptions we declare it by using throws keyword at method signature. **Example:-** IO Exception, SQL Exception

**UnChecked Exceptions:-** The exceptions which occur at run time is known as unchecked exceptions. **Example:-** NullPointerException, ArithematicEception.

1. **How do you create custom Exception?**

Custom Exceptions are those which we declare inside method ie.. we explicity throw an exception by using throw keyword.

1. **How does Exception propagation works?**

When an exception is thrown from top of stack and if it is not caught, it drops down the call stack to previous method, and if the exception is not caught there too then it further drops down to previous method. This process continuous until the exception caught or they reach very bottom of call stack.

1. **Difference b/w throw and throws?**

Both throw and throws are keyword in java and are used to handle an exception.

Throws:- If there is any exception then we declare an exception by using throws keyword. It is declared in method signature and followed by a class. We can declare multiple exceptions (IOException, SQLException are declared by using throws.)

Throw:- We explicitly throw an exception. Generally these are custom exception. Where we declare them within the method body, it is followed by an instance. You cannot throw multiple exceptions.

1. **Exception vs Error ?**

Exception:-

* Exceptions are mainly caused by application itself.
* Exceptions can be both checked and unchecked exceptions.
* As exceptions are recoverable it can be handled by using some keywords like try, catch, throw, throws.
* Checked exception:- IOException, SQLException

Unchecked Exceptions:- NullPointerException, ArithmeticException.

Error:-

* Error are caused by environment in which application is running.
* All errors in java are unchecked exceptions.
* It is impossible to recover from error.
* Some of the error can be StackOverFlowError, OutOfMemoryError.

1. **ArrayList vs LinkedList**

Array List and Linked List implements List Interface. If we want to represent group of objects as single unit where insertion order is maintained and duplicates are allowed we can use anything either Array List or Linked List but there are significant differences between them like Array List is index based and Linked List is node based.

Array List:-

* Internally it uses dynamic array, if any element is removed from an array all the bits are shifted in memory. So, manipulation is slow.
* Array List is preferred if our operation is based on searching an element as Array List implements Random-access.(either first element or last element it finds with same speed).

Linked List:-

* Linked List uses doubly linked list to store elements. If any element is removed from the list then no bit shifting is required in memory. So, manipulation is fast. So, Linked List is preferred if our operation is for adding or removing basically I could say for manipulation.

1. **List vs Set:-**

* List and set are interfaces which extends collection interface.

**List**:-If we want to represent group of objects as single unit where insertion order is maintained and duplicates are allowed we use list interface

Array List, Linked List, Vector implements List Interface.

**Set:-**If we want to represent group of objects as single unit where insertion order is not maintained and duplicates are not allowed we use Set Interface. But there are few implementations in set which maintains the insertion order is LinkedHashSet.

We know that List allow duplicates while Set doesn’t allow if you are trying to insert a duplicate element then old value is replaced by new value.

1. **HashSet, LinkedHashSet, TreeSet implements Set interface**. **Which one you prefer beween ArrayList vs LinkedList?**

**LinkedList** -allows constant-time insertions or removals using iterators but only sequential access of elements. In other words you can walk through elements using forward or backward directions. In order to find an element it takes a lot of time it is almost proportional to size of elements in the list.

**ArraryList**:- On other hand in Array List we can find any element with same speed either first or last element we can find it with same speed because Array List implements Random-access. While in Insertion or Removal of elements bits should be shifted in memory, and also If you add more elements than capacity, a new array list is created and old array is copied and new elements will be added.

1. What is JDBC API and when do we use it?

JDBC is a Java API that is used to get the database connection, run SQL queries and stored procedures in the database server and process the results. JDBC API uses jdbc drivers to connects to the database. It allows us to work with relational databases. We can use JDBC API

JDBC API is written in a way to allow loose coupling between our Java program and actual JDBC drivers that makes our life easier in switching from one database to another database servers easily.