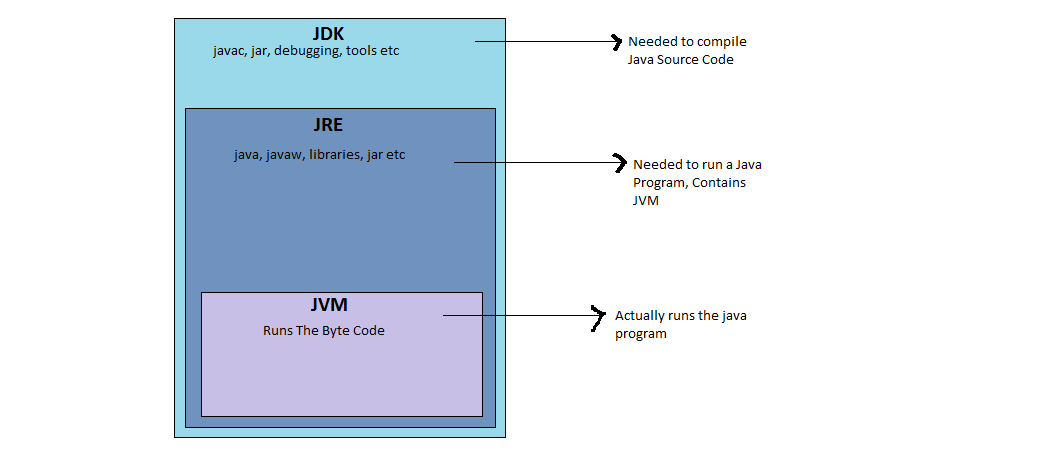
**JAVA Interview Questions**

**Q> what is JDK , JRE and JVM**(VM ware,IBM, HCL)

Ans:-



**Q> What is wrapper class and why do we need them or where do we use it (**VM Ware,symphony telca**)**

Ans:-Java is an object-oriented language and can view everything as an object. A simple file can be treated as an object , an address of a system can be seen as an object , an image can be treated as an object (with java.awt.Image) and a simple data type can be converted into an object (with wrapper classes). Wrapper classes are used to convert any data type into an object.

The primitive data types are not objects; they do not belong to any class; they are defined in the language itself. Sometimes, it is required to convert data types into objects in Java language. For example, upto JDK1.4, the data structures accept only objects to store. A data type is to be converted into an object and then added to a Stack or Vector etc. For this conversion, the designers introduced wrapper classes.

**What are Wrapper classes?**

As the name says, a wrapper class wraps (encloses) around a data type and gives it an object appearance. Wherever, the data type is required as an object, this object can be used. Wrapper classes include methods to unwrap the object and give back the data type. It can be compared with a chocolate. The manufacturer wraps the chocolate with some foil or paper to prevent from pollution. The user takes the chocolate, removes and throws the wrapper and eats it.

Observe the following conversion.

int k = 100;

Integer it1 = new Integer(k);

The int data type k is converted into an object, it1 using Integer class. The it1 object can be used in Java programming wherever k is required an object.

The following code can be used to unwrap (getting back int from Integer object) the object it1.

int m = it1.intValue();

System.out.println(m\*m); // prints 10000

intValue() is a method of Integer class that returns an int data type.

**Importance of Wrapper classes**

There are mainly two uses with wrapper classes.

**1)** To convert simple data types into objects, that is, to give object form to a data type; here constructors are used.

**2)** To convert strings into data types (known as parsing operations), here methods of type parseXXX() are used.

**Features of the Java wrapper Classes.**

**1)** Wrapper classes convert numeric strings into numeric values.

**2)** The way to store primitive data in an object.

**3)** The valueOf() method is available in all wrapper classes except Character

**4)** All wrapper classes have typeValue() method. This method returns the value of the object as its primitive type.

**Q> javaAutoboxing and unboxing**(HSBC)

Ans:- The automatic conversion of primitive data types into its equivalent Wrapper type is known as boxing and opposite operation is known as unboxing. This is the new feature of Java5. So java programmer doesn't need to write the conversion code.

**Example:-**

**Q> what thing goes to heap and stack**(sears,synechron,Sigma Edge)

**Stack :- faster**

**Heap :- slow**

**All premetive data types and functions calls are stored on stack.**

**If object is created then only its reference is created on stack.**

**All non premetive data types such as string, arrays and objects are stored on heap**

**Q> what is the difference between hasmap and hastable**(wipro, J & J, Talentica)

**Ans :-** There are several differences between HashMap and Hashtable in Java:

Hashtable is synchronized, whereas HashMap is not. This makes HashMap better for non-threaded applications, as unsynchronized Objects typically perform better than synchronized ones.

Hashtable does not allow null keys or values. HashMap allows one null key and any number of null values.

HashMap is not an ordered collection which means it does not return the keys and values in the same order in which they have been inserted into the HashMap

Code for HashTable

**publicclass**HashTableExample {

**publicstaticvoid**main(String args[]){

Hashtable<Integer,String>ht=**new**Hashtable<Integer,String>();

ht.put(100,"pune");

ht.put(101,"mumbai");

System.***out***.println(ht.get(100));

System.***out***.println(ht.get(101));

}

}

**Q> What is meant by pass by reference and pass by value in Java?**(HSBC, IBM, Syntel)

Ans:- Pass by reference means, passing the address itself rather than passing the value. Pass by value means passing a copy of the value.

**Q> What is Byte Code?** (sears,,bmc)

**Or**

**What gives java it’s “write once and run anywhere” nature?**

**or**

**What makes Java Platform independent**(bmc)

Ans:- All Java programs are compiled into class files that contain bytecodes. These byte codes can be run in any platform

and hence java is said to be platform independent.

**Q>Expain the reason for each keyword of public static void main(String args[])?** (infosys, CTS, PTC, Avaya, Wipro, Amdocs)

Ans:- public- main(..) is the first method called by java environment when a program is executed so it has to accessible from java environment. Hence the access specifier has to be public.

static: Java environment should be able to call this method without creating an instance of the class , so this method must be declared as static.

void: main does not return anything so the return type must be void

The argument String indicates the argument type andarg is an array for string given during command line.

**Q> What are the differences between == and .equals() ?**( xpanxion, media ocean, cts, scale arc, synerzip, talentica, Wipro, HCL, TechM, Mphasis, TCS)

**Or**

**what is difference between == and equals**

**Or**

**Difference between == and equals method**

**Or**

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

**Or**

**How is it possible for two String objects with identical values not to be equal under the == operator?**

Ans:-

The == operator compares two objects to determine if they are the same object in memory i.e. present in the same memory location.

It is possible for two String objects to have the same value, but located in different areas of memory.

== compares references while .equals compares contents. The method public booleanequals(Object obj) is provided by the Object class and can be overridden.

The default implementation returns true only if the object is compared with itself, which is equivalent to the equality operator == being used to compare aliases to the object. String,

BitSet, Date, and File override the equals() method. For two String objects, value equality means that they contain the same character sequence. For the Wrapper classes, value equality means that the primitive values are equal.

Example:-

--------

public class EqualsTest {

public static void main(String[] args) {

String s1 = "abc"; //this may or

String s2 = s1;

String s5 = "abc";

String s3 = new String("abc");

String s4 = new String("abc");

System.out.println("== comparison : " + (s1 == s5)); //== comparison : true

System.out.println("== comparison : " + (s1 == s2)); // == comparison : true

System.out.println("Using equals method : " + s1.equals(s2)); // true

System.out.println("== comparison : " + s3 == s4); //== comparison : false

System.out.println("Using equals method : " + s3.equals(s4)); // true

}

}

Output

-----------

== comparison : true

== comparison : true

Using equals method : true

== comparison : false

Using equals method : true

**Note :-**

[String](http://www.coderanch.com/t/410859/java/java/string-stringbuffer-stringbuilder-performance) a = "abc";   
  
May or may not create a new String object.   
If a String object with the literal "abc" already exists the reference 'a' will only point to it.   
Since String objects are immutable.   
  
Where as,   
String a = new String("abc");   
will always create a new Sting object.

**Q> What if the static modifier is removed from the signature of the main method? (xpanxion, synechron, 3i infotech, Aftek )**

**Or**

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

Ans:-

Program compiles. But at runtime throws an error “NoSuchMethodError”.

**Q> What is the difference between final, finally and finalize? What do you understand by the java final keyword? (bmc, infy, hsbc)**

**Or**

**What is final, finalize() and finally?**

**Or**

**What is finalize() method?**

**Or**

**What is the difference between final, finally and finalize?**

**Or**

**What does it mean that a class or member is final?**

Ans:-

final – declare constant

finally – handles exception

finalize – helps in garbage collection

**Q>Describe the principles of OOPS** ( synerzip, entity data, CTS, persistent, caterpillar, j&J, synchrony, sunguard, cybage, bitwise )

Ans:- There are three main principals of oops which are called Polymorphism, Inheritance and Encapsulation.

**Q>Explain the Polymorphism principle. Explain the different forms of Polymorphism.**( all the companies )

Ans:- Polymorphism in simple terms means one name many forms. Polymorphism enables one entity to be used as a general category for different types of actions. The specific action is determined by the exact nature of the situation.

Polymorphism exists in three distinct forms in Java:

• Method overloading

• Method overriding through inheritance

• Method overriding through the Java interface

**Q> In System.out.println(), what is System, out and println?( hsbc, sears, xpanxion, cybage)**

Ans:- System is a predefined final class, out is a PrintStream object and println is a built-in overloaded method in the out object.

**Q> What are Java Access Specifiers?**(allSccripts, talentica,Cybage)

**Or**

**What is the difference between public, private, protected and default Access Specifiers?**

**Or**

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

Ans:- Access specifiers are keywords that determine the type of access to the member of a class. These keywords are for allowing

privileges to parts of a program such as functions and variables. These are:

• Public : accessible to all classes

• Protected : accessible to the classes within the same package and any subclasses.

• Private : accessible only to the class to which they belong

• Default : accessible to the class to which they belong and to subclasses within the same package

**Q> Which class is the superclass of every class? (**Symantec**)**

Ans:- Object.

**Q> What is the difference between static and non-static variables?**(all the companies)

**Or**

**What are class variables?**

**Or**

**What is static in java?**

**Or**

**What is a static method?**

Ans:-

A static variable is associated with the class as a whole rather than with specific instances of a class. Each object will share a common copy of the static variables i.e. 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. These are declared outside a class and stored in static memory. Class variables are mostly used for constants. Static variables are always called by the class name. This variable is created when the program starts and gets destroyed when the programs stops. The scope of the class variable is same an instance variable. Its initial value is same as instance variable and gets a default value when its not initialized corresponding to the data type. Similarly, a static method is a method that belongs to the class rather than any object of the class and doesn’t apply to an object or even require that any objects of the class have been instantiated.

Static methods are implicitly final, because overriding is done based on the type of the object, and static methods are attached to a class, not an object. A static method in a superclass can be shadowed by another static method in a subclass, as long as the original method was not declared final. However, you can’t override a static method with a non-static method. In other words, you can’t change a static method into an instance method in a subclass.

Non-static variables take on unique values with each object instance.

**Q> Can I have multiple main methods in the same class?**(xorient, cts)

Ans:- We can have multiple overloaded main methods but there can be only one main method with the following signature :

public static void main(String[] args) {}

**Q> Does Java support multiple inheritance?**(all the companies)

Ans :- No java does not support it. but there is a workaround. we can achieve it using interfaces.

**Q> What is constructor?**(KPIT, tech M)

Ans:- Constructor is just like a method that is used to initialize the state of an object. It is invoked at the time of object creation.

**Q> Difference between method overloading and method overriding**(all the companies)

Ans:-

Method Overloading

------------------

1) Method overloading increases the readability of the program.

2) method overlaoding is occurs within the class.

3) In this case, parameter must be different. In this case, parameter must be same.

Method Overriding

------------------

1) Method overriding provides the specific implementation of the method that is already provided by its super class.

2) Method overriding occurs in two classes that have IS-A relationship.

3) In this case, parameter must be same.

**Q>Can we override static method?**(Persistent, synechron)

Ans:- No, you can't override the static method because they are the part of class not object

**Q> What is Runtime Polymorphism?**(Persistent, CTS, techM, Vodafone)

Ans:- Runtime polymorphism or dynamic method dispatch is a process in which a call to an overridden method is resolved at runtime rather than at compile-time.

In this process, an overridden method is called through the reference variable of a super class. The determination of the method to be called is based on the object being referred to by the reference variable.

**Q> What is difference between abstract class and interface?** (all the companies)

Ans:-

Abstract class

--------------

1)An abstract class can have method body (non-abstract methods).

2)An abstract class can have instance variables.

3)An abstract class can have constructor \*\*\*\*\*\*\*\*\*\*\*\*\*\*

4)An abstract class can have static methods.

5)You can extends one abstract class.

Interface

---------

1)Interface have only abstract methods.

2)An interface cannot have instance variables.

3)Interface cannot have constructor.

4)Interface cannot have static methods.

5)You can implement multiple interfaces.

**Q>Does** [**Interfaces allows constructors?**](http://www.instanceofjava.com/2016/02/java-interface-constructor-example.html)

**Ans:-**

* No. Interfaces does not allow constructors.
* Why interface does not have constructor? The variables inside interfaces are static final variables means constants and we can not create object for interface so there is no need of constructor in interface that is the reason interface doesn't allow us to create constructor.

**Q>**[**Can we create static constructor in java**](http://www.instanceofjava.com/2016/02/java-static-constructor-method.html)(thoughtworks)

**Ans:-**

* No. We can not create constructor with static.
* If we try to create a static constructor compile time error will come: Illegal modifier for the constructor

**Q> What is difference between Checked Exception and Unchecked Exception?**(persistent)

Ans:-

1)Checked Exception

The classes that extend Throwable class except RuntimeException and Error are known as checked exceptions e.g.IOException,SQLException etc. Checked exceptions are checked at compile-time.

2)Unchecked Exception

The classes that extend RuntimeException are known as unchecked exceptions e.g. ArithmeticException,NullPointerException etc. Unchecked exceptions are not checked at compile-time.

**Q>Is it necessary that each try block must be followed by a catch block?**(infy, techM,persistent,nuance)

Ans:- It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

**Q>What is finally block?(**infy,cts,techm**)**

Ans:-finally block is a block that is always executed

**Q> What is difference between throw and throws? (**persitent**)**

Ans:-

throw keyword

-------------

1)throw is used to explicitly throw an exception. throws is used to declare an exception.

2)checked exceptions can not be propagated with throw only. Un checked exception can be propagated with throws.

3)throw is followed by an instance. throws is followed by class.

4)throw is used within the method. throws is used with the method signature.

5)You cannot throw multiple exception You can declare multiple exception e.g. public void method()throws IOException,SQLException.

throws keyword

--------------

1)throws is used to declare an exception.

2)checked exception can be propagated with throws.

3)throws is followed by class.

4)throws is used with the method signature.

5)You can declare multiple exception e.g. public void method()throws IOException,SQLException.

**Q> What is the difference between String, StringBuffer and StringBuilder ?**(all the companies)

Ans:-

String:-

------

String is **immutable**  ( once created can not be changed )object . The object created as a String is stored in the Constant String Pool .

Every immutable object in Java is thread safe ,that implies String is also thread safe . String can not be used by two threads simultaneously.

String once assigned can not be changed.

Example:-

String demo = " hello " ;

// The above object is stored in constant string pool and its value can not be modified.

demo="Bye" ; //new "Bye" string is created in constant pool and referenced by the demo variable

// "hello" string still exists in string constant pool and its value is not overrided but we lost reference to the "hello"string

**StringBuffer( Thread safe)**

---------------

StringBuffer is mutable means one can change the value of the object . The object created through StringBuffer is stored in the heap .StringBuffer has the same methods as the StringBuilder , but each method in StringBuffer is synchronized that is StringBuffer is thread safe .

Due to this it does not allow two threads to simultaneously access the same method . Each method can be accessed by one thread at a time .

But being thread safe has disadvantages too as the performance of the StringBuffer hits due to thread safe property .Thus StringBuilder is faster than the StringBuffer when calling the same methods of each class.

StringBuffer value can be changed , it means it can be assigned to the new value . Nowadays its a most common interview question ,the differences between the above classes .

String Buffer can be converted to the string by using

toString() method.

Example:-

StringBuffer demo1 = new StringBuffer("Hello") ;

// The above object stored in heap and its value can be changed .

demo1=new StringBuffer("Bye");

// Above statement is right as it modifies the value which is allowed in the StringBuffer

**StringBuilder (Not a thread safe)**

--------------

StringBuilder is same as the StringBuffer , that is it stores the object in heap and it can also be modified . The main difference between the StringBuffer and StringBuilder is that StringBuilderis not thread safe.

StringBuilder is fast as it is not thread safe .

Example:-

StringBuilder demo2= new StringBuilder("Hello");

// The above object too is stored in the heap and its value can be modified

demo2=new StringBuilder("Bye");

// Above statement is right as it modifies the value which is allowed in the StringBuilder

Q> What is the difference between abstraction & encapsulation? (Schlumberger)

Ans :-

|  |  |
| --- | --- |
|  | [**Abstraction**](https://en.wikipedia.org/wiki/Abstraction_%28computer_science%29#Abstraction_in_object_oriented_programming) occurs during class level design, with the objective of **hiding** the implementation complexity of **how** the the features offered by an API / design / system were implemented, in a sense simplifying the 'interface' to access the underlying implementation.  For example, a Java developer can make use of the high level features of [FileInputStream](https://docs.oracle.com/javase/7/docs/api/java/io/FileInputStream.html) without concern for how it works (i.e. file handles, file system security checks, memory allocation and buffering will be managed internally, and are hidden from consumers). This allows the implementation of FileInputStream to be changed, and as long as the API (interface) to FileInputStream remains consistent, code built against previous versions will still work.  Similarly, when designing your classes, you will want to hide internal implementation from others as far as possible.  [**Encapsulation**](https://en.wikipedia.org/wiki/Information_hiding), often referred to as "Information Hiding", revolves more specifically around the hiding of the internal data (e.g. state) owned by a class instance, and enforcing access to the internal data in a controlled manner.  For example, class fields can be made private by default, and only if external access to these was required, would a get() and/or set() (or Property) be exposed from the class. (In modern day OO languages, fields can be marked as readonly / final / immutable which further restricts change, even within the class). |

**Q> What is static block**

Ans :- Static block is used for initializing the static variables.This block gets executed when the class is loaded in the memory. A class can have multiple Static blocks, which will execute in the same sequence in which they have been written into the program.

classJavaExample{

staticintnum;

staticStringmystr;

static

{

num = 97;

mystr = "Static keyword in Java";

}

publicstaticvoidmain(Stringargs[])

{

System.out.println("Value of num: "+num);

System.out.println("Value of mystr: "+mystr);

}

}

**Singleton class**

**Synchronisation**

**String Pool**

**Retry Analyser**