1. **List of Languages that can be executed in JVM.**

Groovy: Programming and scripting Languages

Scala : Object oriented and functional programming lanagugae

Clojure: Programming language

Kotlin: Object oriented language

Ceylon: Object Oriented language

1. **Ques: difference between JVM and JRE**

Ans: JVM is part of JRE, JRE includes both JVM and API.

JVM includes Class loaders, runtime areas and execution engine.

A Java Runtime Environment(JRE) is a prerequisite for Running Java Application on any computer.

1. **What is the difference between JDK and JVM?**

Java Development Kit (JDK) is for development purpose and JVM is a part of it to execute the java programs.

JDK provides all the tools, executables and binaries required to compile, debug and execute a Java Program. The execution part is handled by JVM to provide machine independence.

1. **What is the difference between JDK and JRE**

A JRE contains a (JVM)Java Virtual Machine, all standard, Core java Classes and Runtime Libraries.It does not contain any development Tools such as Compiler,Debugger, etc.. JDK(Java Development Kit) is a whole Package required to Java Development which essential contails JRE+JVM and Tools required to compile and Debug,Execute Java Applications.

1. **Ques: components of JDK.**

Ans: javac, java, javap, javah,javadoc, jdb, appletviewer

1. **Ques : what comprises java platform**

Ans: JVM and java API together form java platform. Java platform is your java runtime environment.

1. **Ques: what is java class file magic number.**

Ans: A Magic Number of a class file a unique Identifier for Tools to quickly differentiate class files from non class files.The First Four Bytes of each Java Class file has th Magic Value as 0xCAFEBABE.

A Magic Number of a class file is a unique identifier for tools to quickly differentiate class files from non class files.The first four bytes of each Java class file has the magic value as 0xCAFEBABE.And the answer to why this number,I do not actually know but there may be very few sensible and acceptable options possible constructed from letters A-F which can surely not be 'CAFEFACE' or 'FADECAFE'....

1. **Responsibilities and type of class loader**

Ans: loading, linking and initialization. Bootstrap and user define class loaders.

### Which class is the superclass of all classes?

java.lang.Object is the root class for all the java classes and we don’t need to extend it.

### Why Java is not pure Object Oriented language?

Java is not said to be pure object oriented because it support primitive types such as int, byte, short, long etc. I believe it brings simplicity to the language while writing our code. Obviously java could have wrapper objects for the primitive types but just for the representation, they would not have provided any benefit.

As we know, for all the primitive types we have wrapper classes such as Integer, Long etc that provides some additional methods.

There are seven qualities to be satisfied for a programming language to be pure Object Oriented. They are:

1. Encapsulation/Data Hiding

2. Inheritance

3. Polymorphism

4. Abstraction

5. All predifined types are objects

6. All operations are performed by sending messages to objects

7. All user defined types are objects.

If you look at these seven qualities, Java does satisfy most of them. Java supports Encapsulation at class and package level, It supports Abstraction, Inheritance and Polymorphism, and all user defined types are also objects. What it doesn't support is #5, all predefined types are not objects in Java, because you can define primitive types. This means it also violates #6. That's why Java is not a pure object oriented language.

Smalltalk is purely object oriented language, here even primitive types are primitive classes.

On closing notes, You can make your program pure object oriented by using [Autoboxing](http://java67.blogspot.sg/2012/09/autoboxing-enum-generics-varargs-java-5-features.html), but Java compiler supports primitive data types, so Java cannot be Pure object oriented unless it makes everything in terms of objects. In short, Java is not pure object oriented programming language because it supports primitive data types and everything is not object in Java.

One more reason, java does not support all types of inheritance through classes only. It supports multiple inheritance through one class and multiple interfaces.

### What is the importance of main method in Java?

main() method is the entry point of any standalone java application. The syntax of main method is : public static void main(String args[]).

main method is public and static so that java can access it without initializing the class. The input parameter is an array of String through which we can pass runtime arguments to the java program. Check this post to learn [how to compile and run java program](http://www.journaldev.com/481/java-tutorial-2-writing-and-running-first-java-program).

1. **Can we overload main method?**

Yes, we can have multiple methods with name “main” in a single class. However if we run the class, java runtime environment will look for main method with syntax as public static void main(String args[]).

1. **Can we have multiple public classes in a java source file?**

We can’t have more than one public class in a single java source file. A single source file can have multiple classes that are not public.

1. **What is Java Package and which package is imported by default?**

Java package is the mechanism to organize the java classes by grouping them. The grouping logic can be based on functionality or modules based. A java class fully classified name contains package and class name. For example, java.lang.Object is the fully classified name of Object class that is part ofjava.lang package. java.lang package is imported by default and we don’t need to import any class from this package explicitly.

### What is JVM and is it platform independent?

Java Virtual Machine (JVM) is the heart of java programming language. JVM is responsible for converting byte code into machine readable code. JVM is not platform independent, thats why you have different JVM for different operating systems. We can customize JVM with Java Options, such as allocating minimum and maximum memory to JVM. It’s called virtual because it provides an interface that doesn’t depend on the underlying OS.

1. **Why c++ is called partial or semi object oriented?**

Sol: c++ supports primitive datatype

In C++ main function is written outside the class, which means you can execute the program without creating a class.

A language is said to be object oriented language if it works on classes and objects.

1. **Components of sandbox security model ?**
   1. Class loader architecture
   2. Security features built in JVM
   3. Security Manager
2. [**What is heap and stack?**](http://interviewjava.blogspot.com/2007/04/what-is-heap-and-stack.html)  
   The heap is the part of memory of JVM where all objects reside.The stack is consisted of stack frames.When a thread invokes a method,the JVM pushes a new frame onto that thread's Java stack.Each stack frame is consisted of operand stack and the local variable array.All arguments,local variables,intermediate computations and return values if any are kept in these stack corresponding to the method invoked.The stack frame on the top of the stack is called the active stack frame,which is the current place of execution.When the method completes, the virtual machine pops and discards the frame for that method.

Difference between Heap and stack

The major difference between Heap and Stack memory are:

|  |  |  |
| --- | --- | --- |
| **Features** | **Stack** | **Heap** |
| ***Memory*** | Stack memory is used only by one thread of execution. | Heap memory is used by all the parts of the application. |
| ***Access*** | Stack memory can’t be accessed by other threads. | Objects stored in the heap are globally accessible. |
| ***Memory Management*** | Follows LIFO manner to free memory. | Memory management is based on the generation associated with each object. |
| ***Lifetime*** | Exists until the end of execution of the thread. | Heap memory lives from the start till the end of application execution. |
| ***Usage*** | Stack memory only contains local primitive and reference variables to objects in heap space. | Whenever an object is created, it’s always stored in the Heap space. |

1. [**How JVM performs Garbage Collection?**](http://interviewjava.blogspot.com/2007/04/how-jvm-performs-garbage-collection.html) **Whenit is done?**

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

A Java object is subject to garbage collection when it becomes unreachable to the program in which it is used.

Whenever a reference to an object on heap lies dangling or no longer in use then it becomes eligible for being garbage collected by JVM.JVM specifications do not force any specific kind of garbage collection algorithm though there are several algorithms like reference counting,tracing,compacting,copying,generational etc. in place.It is very important that garbage collection should be efficient and non interfering in execution of Java programs.There is a trade off between ease of implementation versus better performance while implementing garbage collection feature for a JVM.

1. .[**What is the difference between interpreted code and compiled code?**](http://interviewjava.blogspot.com/2007/04/what-is-difference-between-interpreted.html)  
   An interpreter produces a result from a program, while a compiler produces a program written in assembly language and in case of Java from bytecodes.The scripting languages like JavaScript,Python etc. require Interpreter to execute them.So a program written in scripting language will directly be executed with interpreter installed on that computer,if it is absent then this program will not execute.While in case of compiled code,an assembler or a virtual machine in case of Java is required to convert assembly level code or bytecodes into machine level instructions/commands.Generally, interpreted programs are slower than compiled programs, but are easier to debug and revise.
2. **What is difference between compiler and interpreter?**
3. **Arrays are primitive or reference data types in Java?**

Reference data types

1. **Can we use arrays without initialization in java ? yes/no**

No, arrays are objects in java

1. **What are instance variable? Are they need to initialize before use like local variable ?**

Instance variables are those which are defined at the class level. Instance variables need not be initialized before using them as they are automatically initialized to their default values.

1. **Is main required for every class?**

**No**

1. **Why main method is static in java?**

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

1. **Can main method be final? Is there any effect on derive classes?**

Yes, drive class can not have its main.

1. **Does order of public and static matter in main method?**

No, only void should come before main().

1. **List access specifiers that can be attach with external class?**

**Only public and default.**



|  |  |
| --- | --- |
| **Compiler** | **Interpreter** |
| **1** | Compiler Takes **Entire** program as input | Interpreter Takes **Single** instruction as input . |
| **2** | Intermediate Object Code is**Generated** | **No** Intermediate Object Code is**Generated** |
| **3** | Conditional Control Statements are Executes **faster** | Conditional Control Statements are Executes **slower** |
| **4** | **Memory Requirement** : **More**(Since Object Code is Generated) | **Memory Requirement** is **Less** |
| **5** | Program need not be **compiled** every time | Every time higher level program is converted into lower level program |
| **6** | **Errors** are displayed after **entire program** is checked | **Errors** are displayed for **every instruction** interpreted (if any) |
| **7** | **Example** : C Compiler | **Example** : BASIC |

|  |  |
| --- | --- |
| Interpreter | Compiler |
| Translates program one statement at a time. | Scans the entire program and translates it as a whole into machine code. |
| It takes less amount of time to analyze the source code but the overall execution time is slower. | It takes large amount of time to analyze the source code but the overall execution time is comparatively faster. |
| No intermediate object code is generated, hence are memory efficient. | Generates intermediate object code which further requires linking, hence requires more memory. |
| Continues translating the program until the first error is met, in which case it stops. Hence debugging is easy. | It generates the error message only after scanning the whole program. Hence debugging is comparatively hard. |
| Programming language like Python, Ruby use interpreters. | Programming language like C, C++ use compilers. |

Question: Difference between constructor and method ?

A constructor is a member function of a class that is used to create objects of that class. It has the same name as the class itself, has no return type, and is invoked using the new operator.

A method is an ordinary member function of a class. It has its own name, a return type (which may be void), and is invoked using the dot operator.