

Java Viva Question and Answer

Q. Why Java?

A. The programs that we are writing are very similar to their counterparts in several other languages, so our choice of language is not crucial. We use Java because it is widely available, embraces a full set of modern abstractions, and has a variety of automatic checks for mistakes in programs, so it is suitable for learning to program. There is no perfect language, and you certainly will be programming in other languages in the future.

Q. Do I really have to type in the programs in the book to try them out?

A. Everyone should type in `HelloWorld.java`, but you can find all of the code in this book (and much more) on this booksite.

Q. What are Java's rules regarding tabs, spaces and newline characters?

A. There are not many. Java compilers treat them all to be equivalent. For example, we could also write `HelloWorld` as follows:

```
public class HelloWorld { public static void main (
String [] args) { System.out.println("Hello World") ; } }
```

But we do normally adhere to spacing and indenting conventions when we write Java programs, just as we always indent paragraphs and lines consistently when we write prose or poetry.

Q. What are the rules regarding quotation marks?

A. Material inside quotation marks is an exception to the rule of the previous question: things within quotes are taken literally so that you can precisely specify what gets printed. If you put any number of successive spaces within the quotes, you get that number of spaces in the output. If you accidentally omit a quotation mark, the compiler may get very confused, because it needs that mark to distinguish between characters in the string and other parts of the program. To print a quotation mark, a newline, or a tab, use `\`, `\n`, or `\t`, respectively, within the quotation marks.

Q. What is the meaning of the words `public`, `static` and `void`?

A. These keywords specify certain properties of `main()` that you will learn about later in the book. For the moment, we just include these keywords in the code (because they are required) but do not refer to them in the text.

Q. What happens when you omit a brace or misspell one of the words, like `public` or `static` or `main`? **A.** It depends upon precisely what you do. Such errors are called *syntax errors*. Try it out and see.

Q. Can a program use more than one command-line argument?

Java Viva Question and Answer

A. Yes, you can put several, though we normally use only a few. You refer to the second one as `args[1]`, the third one as `args[2]`, and so forth. Note that we start counting from 0 in Java.

Q. What Java systems libraries and methods are available for me to use?

A. There are thousands of them, but we introduce them to you in a deliberate fashion in this book to avoid overwhelming you with choices.

Q. How should I format my code? How should I comment my code?

A. Programmers use coding guidelines to make programs easier to read, understand, and maintain. As you gain experience, you will develop a coding style, just as you develop style when writing prose. [Appendix B](#) provides some guidelines for formatting and commenting your code. We recommend returning to this appendix after you've written a few programs.

Q. What exactly is a `.class` file?

A. It's a binary file (sequence of 0s and 1s). If you are using Unix or OS X, you can examine its contents by typing `od -x HelloWorld.class` at the command prompt. This displays [the results](#) in hexadecimal (base 16). In deference to Java's name, the first word of every `.class` file is `cafe`.

Q. How do I get the `|` symbol on my keyboard?

A. It's there. Often it's above the `\` symbol.

Q. Java prints out a ton of digits when I `System.out.println()` a double. How can I format it so it displays only 3 digits after the decimal place?

A. Use the method `System.out.printf()` described in Section 1.5.

Q. Why does the integer quotient `-0/3` yield 0, but the double quotient `-0.0/3.0` yields `-0.0`?

A. Java represent integers using something called *two's complement notation*, and there is only one representation of 0. (You'll learn about this in Section 5.1.) Java represents doubles using IEEE specifications, and there are two distinct representations of the number zero, 0 and -0. (You'll learn about this in Section 9.1.)

Q. What happens when I use `/` and `%` with a negative numerator?

A. Try it and see. `-47 / 5 = -9` and `-47 % 5 = -2`. The quotient is always rounded toward zero. To ensure the *Euclidean property* `b * (a / b) + (a % b) = a`, the result of the remainder operator can be negative. This convention was inherited from ancestral

Java Viva Question and Answer

languages like FORTRAN and C. Some languages (but not Java) include both remainder and modulo operators because it is often convenient to have a version that returns only nonnegative integers.

Q. Can I use % with real numbers?

A. Yes. If `angle` is nonnegative, then `angle % (2 * Math.PI)` converts the angle to be between 0 and 2π .

Q. How do I print a " ?

A. Since " is a special character when dealing with strings, you must *escape* the convention rules by using \". For example, `System.out.println("The pig said \"Oink Oink\" afterwards");`.

Q. OK, so then how do I print a \ ?

A. Use "\\".

Q. Are there any restrictions on the variable names I can use?

A. Yes. A [Java identifier](#) is a Java letter, followed by an unlimited sequence of Java letters and Java digits. Java letters and digits can be drawn from the entire Unicode character set, although we only use English language characters in this book. By convention, variables usually begin with a lowercase letter. An identifier cannot be named any of the following reserved words.

abstract	default	goto	package	this
assert	do	if	private	throw
boolean	double	implement	protected	throws
break	else	import	public	transient
byte	enum	instanceof	return	true
case	extends	int	short	try
catch	false	interface	static	void
char	final	long	strictfp	volatile
class	finally	native	super	while
const	float	new	switch	
continue	for	null	synchronized	

Q. What are the precedence rules for operators in Java?

A. This table provides the [precedence order of operators in Java](#).

Q. Is there any difference between `a += b` and `a = a + b`, where `a` and `b` are primitive types?

A. Possibly, if `a` and `b` are of different types. The assignment statement `a += b` is equivalent to `a = (int) (a + b)` if `a` is of type `int`. Thus, if `b` is of type `double`, `a += b` is legal, but `a = a + b` is a compile-time error.

Java Viva Question and Answer

Q. Why do I need to declare the type of a variable in Java?

A. By specifying the type, the compiler can alert you of potential errors, say if you try to multiply an integer with a string. For the same reason, when doing physics calculations, it is always a good idea to keep track of the units and make sure they "type check." For small programs, this may not seem important; for large programs it is crucial. The [Ariane 5 rocket](#) exploded 40 seconds after takeoff because of a bug in its software that incorrectly converted a 64 bit real number into a 16 bit integer.

Q. Why is the type for real numbers called `double`?

A. Historically, the type for floating point numbers was `float`, but they had limited accuracy. The type `double` was introduced as a floating point type with twice as much accuracy.

Q. Is it correct to say that using parentheses can only change

A. Almost, with one surprising exception. The literal value 2147483648 (2^{31}) is only permitted as an operand of the unary minus operator, i.e., -2147483648. Enclosing it in parentheses, i.e., -(2147483648), leads to a compile-time error.

Q. My program is stuck in an infinite loop. How do I stop it?

A. For DrJava, click the menu option *Tools -> Reset Interactions*. For OS X Terminal and Linux shell, type `<ctrl-c>`. That is, hold down the key labeled `ctrl` or `control` and press the `c` key. For Windows Command Prompt type `<ctrl-z>`.

Q. How can I check whether two strings are equal? Using `==` doesn't work.

A. This is one of the difference between primitive types (`int`, `double`, `boolean`) and reference types (`String`). We'll learn about testing strings for equality in Section 3.1.

Q. Why doesn't the statement `if (a <= b <= c)` work?

A. The `<=` operator cannot be chained. Instead, use `if (a <= b && b <= c)`.

Q. Is there an example where you cannot remove the curly braces from a one-statement block?

A. The first code fragment is legal (but pointless), while the second is a compile-time error. Technically, the second line in the second fragment is a declaration and not a statement.

```
// legal
for (int i = 0; i <= N; i++) {
    int x = 5;
```

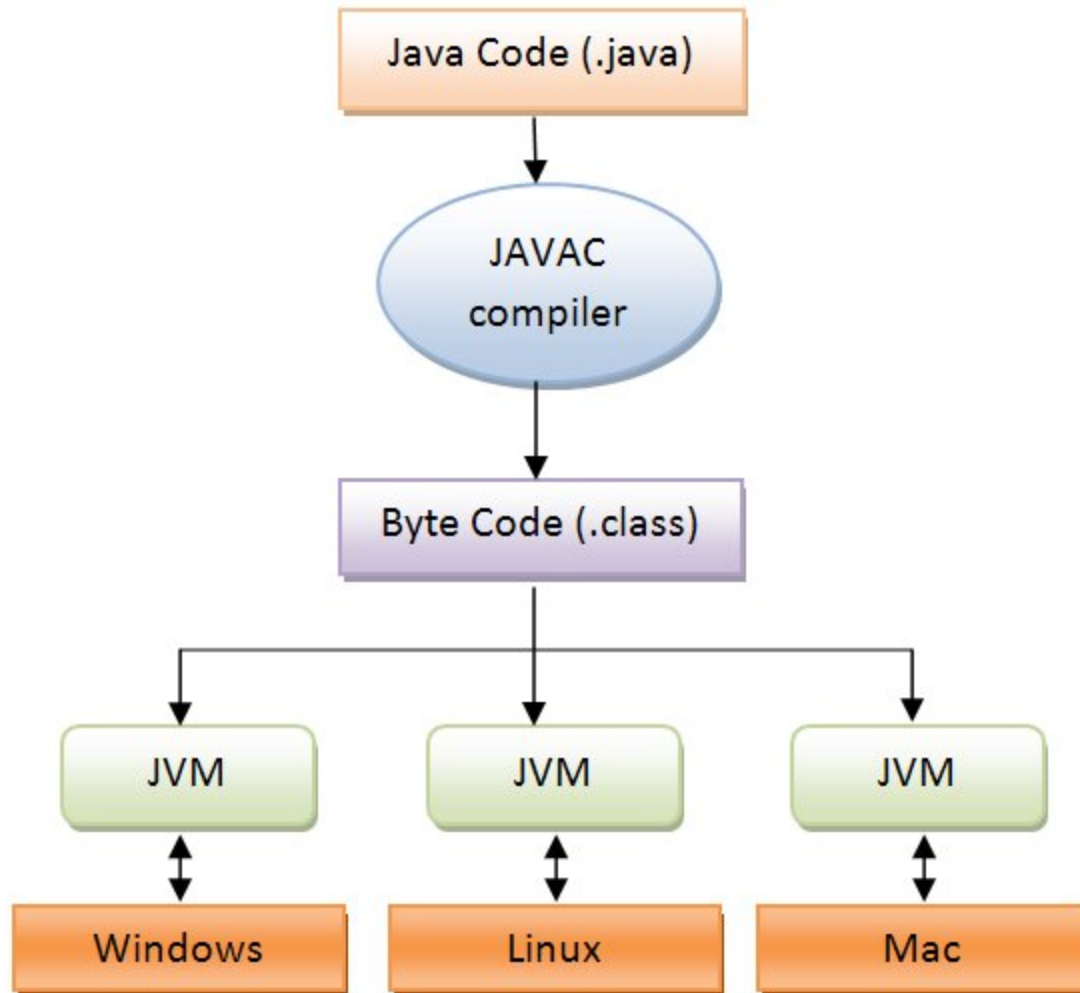
Java Viva Question and Answer

```
}  
  
// illegal  
for (int i = 0; i <= N; i++)  
    int x = 5;
```

Java Virtual Machines

The **JVM** is a crucial component of the Java Platform. Because JVMs are available for many hardware and software platforms, Java can be both middleware and a platform in its own right — hence the trademark write once, run anywhere. The use of the same bytecode for all platforms allows Java to be described as "**compile once, run anywhere**", as opposed to "write once, compile anywhere", which describes cross-platform compiled languages. The JVM also enables such features as Automated Exception Handling that provides 'root-cause' debugging information for every software error (exception) independent of the source code.

Java Viva Question and Answer



The **JVM** is distributed along with a set of standard class libraries that implement the Java API (Application Programming Interface). An application programming interface is what a computer system, library or application provides in order to allow data exchange between them. They are bundled together as the **Java Runtime Environment**.

Java Buzzwords

Java Virtual Machine:-

Java was designed with a concept of 'write once and run everywhere'. Java Virtual Machine plays the central role in this concept. The JVM is the environment in which Java programs execute. It is a software that is implemented on top of real hardware and operating system. When the source

Java Viva Question and Answer

code (.java files) is compiled, it is translated into byte codes and then placed into (.class) files. The JVM executes these bytecodes. So Java byte codes can be thought of as the machine language of the JVM. A JVM can either interpret the bytecode one instruction at a time or the bytecode can be compiled further for the real microprocessor using what is called a just-in-time compiler. The JVM must be implemented on a particular platform before compiled programs can run on that platform.

Object Oriented Programming:-

Since Java is an object oriented programming language it has following features:

- Reusability of Code
- Emphasis on data rather than procedure
- Data is hidden and cannot be accessed by external functions
- Objects can communicate with each other through functions
- New data and functions can be easily added.

Java has powerful features. The following are some of them:-

Simple

Reusable

Portable (Platform Independent)

Distributed

Robust

Secure

High Performance

Dynamic

Java Viva Question and Answer

Threaded

Interpreted

Object Oriented Programming is a method of implementation in which programs are organized as cooperative collection of objects, each of which represents an instance of a class, and whose classes are all members of a hierarchy of classes united via inheritance relationships.

OOP Concepts

Four principles of Object Oriented Programming are

Abstraction

Encapsulation

Inheritance

Polymorphism

Abstraction

Abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of objects and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer.

Encapsulation

Encapsulation is the process of compartmentalizing the elements of an abstraction that constitute its structure and behavior ; encapsulation serves to separate the contractual interface of an abstraction and its implementation.

Encapsulation

- * Hides the implementation details of a class.
- * Forces the user to use an interface to access data
- * Makes the code more maintainable.

Inheritance

Java Viva Question and Answer

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

Polymorphism :-

Polymorphism is the existence of the classes or methods in different forms or single name denoting different implementations.

Java is Distributed

With extensive set of routines to handle TCP/IP protocols like HTTP and FTP java can open and access the objects across net via URLs.

Java is Multithreaded

One of the powerful aspects of the Java language is that it allows multiple threads of execution to run concurrently within the same program. A single Java program can have many different threads executing independently and continuously. Multiple Java applets can run on the browser at the same time sharing the CPU time.

Java is Secure

Java was designed to allow secure execution of code across network. To make Java secure many of the features of C and C++ were eliminated. Java does not use Pointers. Java programs cannot access arbitrary addresses in memory.

Java Viva Question and Answer

NOTE:

Why do we need public static void main(String args[]) method in Java
We need

- public: The method can be accessed outside the class /package
- static: You need not have an instance of the class to access the method
- void: Your application need not return a value, as the JVM launcher would return the value when it exits
- main(): This is the entry point for the application

Garbage collection

Automatic garbage collection is another great feature of Java with which it prevents inadvertent corruption of memory. Similar to C++, Java has a new operator to allocate memory on the heap for a new object. But it does not use delete operator to free the memory as it is done in C++ to free the memory if the object is no longer needed. It is done automatically with garbage collector.

Java Applications

Java has evolved from a simple language providing interactive dynamic content for web pages to a predominant enterprise-enabled programming language suitable for developing significant and critical applications. Today, It is used for many types of applications including Web based applications, Financial applications, Gaming applications, embedded systems, Distributed enterprise applications, mobile applications, Image processors, desktop applications and many more.

Java Viva Question and Answer

If the main() was not static, you would require an instance of the class in order to execute the _____ method.

If this is the case, what would create the instance of the class? What if your class did not have a public constructor?