



# Certified Core Java Developer

## VS-1036

**V-Skills Certifications**

**A Government of India  
&  
Government of NCT Delhi Initiative**

**V-Skills**



## 1. LANGUAGE FUNDAMENTALS

---

The Java language's programming paradigm is implementation and improvement of Object Oriented Programming (OOP) concepts. The Java language has its own rules, syntax, structure and programming paradigms. Today Java had expanded its scope to almost every digital field either it's a mobile device, or a television, or a washing machine, etc. Today Java is used to deliver highly interactive and secure Desktop Applications, Web Based Content, Mobile Applications and Games, and Enterprise Software's. Java helps a programmer or an organization to develop or deploy exciting services and applications.

Since Java is derivative of C++ or C Language, most of its features and syntax are similar in Java except that Java doesn't have concept of pointers. Some of them are variable declaration before they are used and the way both languages declare a variable, use of block codes, structure of loops and conditional statements etc.

### 1.1 About Java Compiler and JVM

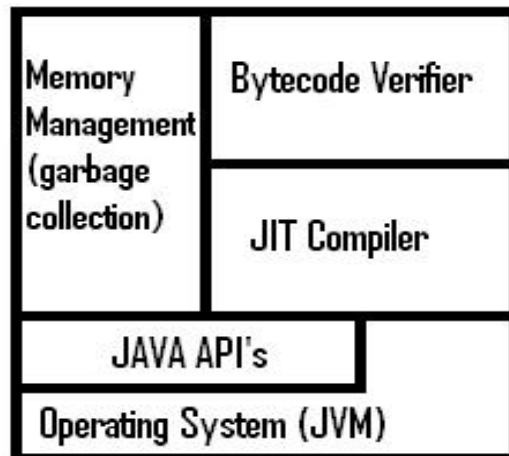
Java compiler is way different than C or C++ compiler. The C or C++ compiler compiles the complete code into machine code, which is dependent on OS. Thus a compiled code of C or C++ in Windows environment can't run on Linux/Unix/Solaris. Unlike C or C++, Java provides cross platform portability. Thus a code written and compiled in Windows environment can be run on Linux/Unix/Solaris/Mac OS environment without any extra efforts. But how is this possible? The answer to this question is very simple; it's because of the Java Compiler. The Java compiler compiles the source code file (.java) into a Bytecode file which has an extension of .class or .jar. This .class or .jar file, i.e. the Bytecode file, is independent of the OS and it can only be understood by JVM (i.e. Java Virtual Machine). This makes Java code portable. The JVM is platform dependent. So the JVM provides an environment to run the .class or .jar file, i.e. the bytecode runs on JVM.



Fig 1.1 How Java Source code converts to Machine **CODE**

### 1.2 Java Runtime Environment (JRE)

Oracle's (earlier known by Sun Microsystems's) Java execution environment is known as Java Runtime Environment (JRE). The JVM and Java APIs are bundled together to form Java Runtime Environment (JRE). The JRE also has a JIT Compiler. Just In Time (JIT) Compiler converts the bytecode into machine code that can be understood by Operating System. With the help of JIT Compiler, the execution of code becomes faster.



### JRE (Java Runtime Environment)

Fig 1.2 Internal view of JRE

#### 1.3 Garbage Collection in Java

Unlike C or C++ memory allocation and de-allocation process, Java (specifically JVM) automatically performs the allocation of memory from the memory heap and keeps track on the resources and when an object is no longer in use, it de-allocates the memory. This provides a better way of memory management and cleans memory automatically. Thus the over head of programmer to clean up memory reduces in Java. Whenever an object is instantiated, it is allocated memory from memory heap and a daemon thread known as Garbage Collector runs in background to keep track in the object. Before removing any object from memory the garbage collection runs finalize() method of object, so that it can perform any clean up/security process (like closing the database connection). When there is no more space available in Java Heap Memory, and a new object is instantiated, it throws an exception named as OutOfMemoryError. There is no manual way to perform garbage collection in Java. Thus as a programmer you can not clean up the heap memory. Although there are functions like System.gc() or Runtime.gc(), but they don't guarantee cleanup of memory.

#### 1.4 Getting Started with Java

Now let's write our first java. Before starting, check your system for following.

##### Check List

- ✓ Java SE Development Kit 6 (JDK6) or above: In this book, we are going to use JDK version 6. You can use the current version of JDK i.e. 7. If you don't have JDK installed in your system, you can download it from [www.oracle.com](http://www.oracle.com)
- ✓ Text Editor: A simple text editor will work for us. We are going to use Notepad, MS Windows default text editor.

We are now ready to write our first program.

##### 1.4.1 Program 1: Hello World

1. Open up notepad and write following code.

```
class Demof{
```

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

2. Now save this file as “Demo.java”. It must be noted that, the name of file must be similar to the class name having main() method.

### 1.4.2 Setting up Environment Variables

To compile a java code, we first need to set environment variables so that our command line does not throw errors like “javac is not recognized as internal or external command”.

The process to set environment variables for different OS is different. Here we are going to cover the major OS.

**1.4.2.1 WINDOWS XP/7 OS:** Right clicks on “My Computer” and click on “Properties” from popped up menu. In System Property dialog box, switch to “Advance” tab. At the end you will find “Environment Variables” button, click on it a dialog box will appear. Click the “New” button under user variables. Now in this dialog box fill variable NAME as “PATH” and variable VALUE as “C:\Program Files\Java\jdk1.6.0\_31\bin;” or the location of bin folder of java if you have custom installation.

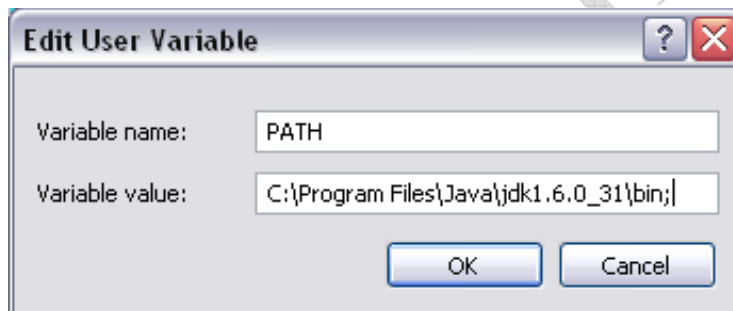


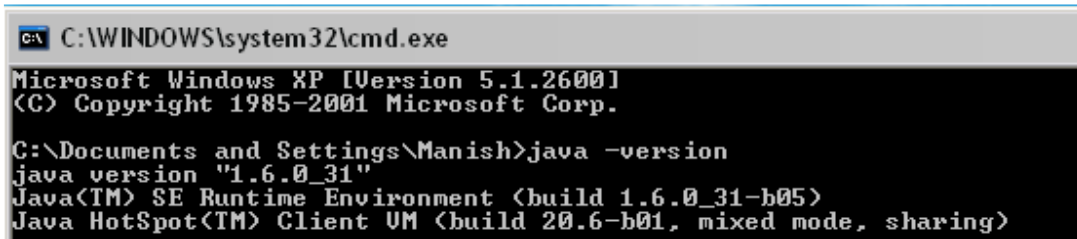
Fig 1.3 Windows Variable Environment Setting Menu

**1.4.2.2 MAC OS:** Download the JDK from Apple’s App Center. Mount the .dmg disk and install the package by double clicking .pkg file. Now open up the terminal from “Applications -> Utility -> Terminal” and type the following command “export JAVA\_HOME=/System/Library/Frameworks/JavaVM.framework/Home”. You need to logout the session and start again to make the changes.

**1.4.2.3 LINUX OS:** In Linux, java is installed to /usr/java by default. If you had a custom installations then change to that directory. Now go to /etc/profile.d directory and create a file name as java.sh. Now write the following content into it.

```
#!/bin/bash
JAVA_HOME=/usr/java/j2se
JAVA_FONTS=/usr/share/fonts/truetype
ANT_HOME=/usr/share/ant
PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$PATH
export PATH JAVA_HOME JAVA_FONTS ANT_HOME
export CLASSPATH=.
```

After setting environment variables, go to command line and type following command “java -version”. If it lists your java version then you are ready to go, otherwise check environment variables again.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Manish>java -version
java version "1.6.0_31"
Java(TM) SE Runtime Environment (build 1.6.0_31-b05)
Java HotSpot(TM) Client VM (build 20.6-b01, mixed mode, sharing)
```

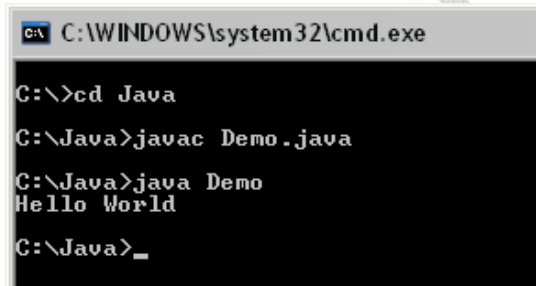
Fig 1.4 Output in Command Line after setting environment

### 1.4.3 Compiling and Executing Program

Step by step instructions to compile the Java program.

- ✓ Open Command prompt and change to the directory in which you have saved .java file, here we have Demo.java file.
- ✓ Now to compile the source code use command “javac Demo.java”. It will generate a .class file.
- ✓ To run the code now type “java Demo”. And you will see the output.

The output of above code will be



```
C:\WINDOWS\system32\cmd.exe

C:\>cd Java
C:\Java>javac Demo.java
C:\Java>java Demo
Hello World
C:\Java>_
```

Fig 1.5 Output of Hello World Program

### 1.4.4 Explanation of Hello World Program

**Class Declaration:** The first line of code is basic way to declare a class. The definition of class resides between the opening and closing curly braces, i.e. {}.

```
class <name> {
}
```

**Main Block:** Every Java program or application executions starts with main() method. The signature of the main() method is :

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

The main() method in java accepts the string arguments as an array here named as “args[]” from the command line. These arguments can be used as input to our program and can be simple text, a condition or can be a file name that we want to use in our program (we’ll see later).

In case of no main() method defined in Java program or application we see an error as following:

Exception in thread "main" java.lang.NoClassDefFoundError: Demo.java.

The System.out.println() line will send the output to the print stream which is command prompt in case we care executing it from command prompt.

## 1.5 Access Modifier in Java

The access modifiers are used to define the privilege of variable, methods and classes in Java. The access modifiers help in improving the Security in Java language. Java has three type of access modifier.

- ✓ Private
- ✓ Protected
- ✓ Public
- ✓ No access modifiers, or default access modifier

### 1.5.1 Private Modifier

Whenever a class, a variable or a method is defined as “private”, it can not be accessed by any other class except itself. This means a private variable, or a method can be accessed by only the class they are defined. No other classes or sub-classes, either in same package or different package can access the private class data.

### 1.5.2 Protected Modifier

A class, variable or method defined as protected can be accessed by the class itself, or by the other classes in same package, and by the sub-classes. No classes from other packages can access the data of the protected class or the protected variable or method.

### 1.5.3 Public Modifier

A public class, variable or a method can be accessed from anywhere in java world. The public classes, variable or methods can be accessed by any class, sub-class, or any package.

### 1.5.4 No Access Modifier

The default access modifier is also known as no access modifier. When no access modifier is used, the class, variables or method can be accessed in the same package. But a sub-class or other packages cannot access to those variables or methods.

#### Visibility

Access Modifiers	Same Class	Same package	Sub-Class	Other package
Public	Y	Y	Y	Y
Protected	Y	Y	Y	N
Private	Y	N	N	N
No Modifier	Y	Y	N	N

Table 1: Visibility of Access Modifiers

### Self Assessment Questions

- Q1. What is this platform-independent code of Java called? (ans c)
- A. Source Code
  - B. Virtual Code
  - C. ByteCODE
  - D. Machine CODE
- Q2. What is correct signature of main() method? (ans a, d)
- A. static public void main(String args[])
  - B. void main()
  - C. public void main(String args[])
  - D. public static void main(String args[])
- Q3. Which parameter does the main() method accepts? (ans d)
- A. Integer
  - B. Long
  - C. Boolean
  - D. String
- Q4. JRE is composed of? (ans b)
- A. JVM
  - B. Both a and c
  - C. Java API
  - D. None of the above
- Q5. Which modifier must be used to access a class within the same package only? (ans b)
- A. Public
  - B. No access modifier
  - C. Protected
  - D. Private