
UNIT 1 INTRODUCTION TO JAVA

Structure	Page Nos.
1.0 Introduction	5
1.1 Objectives	5
1.2 Applets and Applications	5
1.3 Java Buzzwords	6
1.4 The Java Platform	8
1.5 Java Libraries	8
1.6 Starting with Java	9
1.7 Summary	10
1.8 Solutions/Answers	11

1.0 INTRODUCTION

Java was conceived by James Gosling, at Sun Microsystems Inc., in 1991. It took 18 months to develop the first working version. This language was initially called “Oak”, but was renamed “Java” in 1995.

The primary motivation of Java was the need for a platform independent language that could be used to create software to be embedded in various consumer electronic devices, such as microwave ovens and remote controls.

Java is a general purpose, current, class-based, object oriented program language. It incorporates many features of C and C ++ but is designed and organised rather differently with a number of feature of C and C ++ debted, and a new feature from other program language.

Java is strongly typed language. This specification clearly distinguishes between the compile time errors that can and must be detected at compile time, and those that occur at run time.

1.1 OBJECTIVES

After going through this Unit you will be able to:

- differentiate between Applets and Applications;
- define the Java Buzzwords;
- understand the Java platform;
- understand the working of the Java Virtual Machine;
- develop some understanding of commonly used Java libraries, and
- how to write, compile and execute a simple Java Program.

1.2 APPLETS AND APPLICATIONS

The differences between them are as follows:

Applets	Applications
1. Applets can be embedded in HTML pages and downloaded over the Internet, or Intranet.	Applications have no special support in HTML for embedding or downloading.

2. Applets can only be executed inside a Java-compatible container, such as a modern Web Browser.	Applications can be executed from the command line with a small booting utility such as javac.exe or Java.exe.
3. Applets execute under strict security limitations that disallow certain operations, such as accessing files or systems services on the user's computer.	Applications have no inherent security restrictions.
4. Applets are the programs written specially for distribution over a network. These programs contain information to be delivered to the world and involve user interaction, for example, order entry form, registration form mailing, etc.	Applications are system level programs i.e., these programs run in the background and don't involve user interaction, for example, server administration, security manager, etc.

Java offers two flavours of programming, **Applets** and **Applications**. They have the following common characteristics:

- 1) Both programs are composed of one or more files with the "CLASS" extension and having machine independent Java Bytecode.
- 2) Both require JVM (Java Virtual Machine) to execute these Bytecodes.

1.3 JAVA BUZZWORDS

Java provides many features which are also known as Java buzzwords. These are:

1) **Simple**

Java was designated to be very simple and easy to learn. The syntax of Java has been kept nearer to C++ so that the usage of Java does not require extensive training programs to be undertaken. However, in Java the infrequently used, complex features of C++ have not been included. Thus, a programmer aware of the various object-oriented concepts can easily develop applications in Java.

2) **Object Oriented**

Simply stated, object oriented design is a technique that focuses design on the data and on the interfaces rather than modularization of the functionalities or the tools used to develop them. Java uses object-oriented concepts as for basis for S/W design. Java provides a clean and efficient object-based development platform.

3) **Robust**

The multi-platform environment of the web places high demand on the reliability of the program to execute on a variety of systems. Thus, high priority has been given to create robust and highly reliable programs in the design of Java.

It provides extensive compile-time checking followed by a second level of run-time checking. The most common problems in programming languages are related to memory management and exception handling.

The memory management model of Java does not allow the creation of pointers. Java has a true array which allows subscript checking to be performed. Thus, Java programmers need not worry about freeing or corrupting memory as the programs cannot overwrite the end of a memory buffer. Also Java has automatic garbage collection once the memory is no longer required.

Situations like File not found, or division by few, are examples of exceptional condition which are not handled properly by traditional programming.

4) **Secure**

Java is designed to be used in networked and distributed environments where security is of paramount importance. Java supports the creation of applications that cannot be invaded from outside. On the other side, Java programs are executed in their own environment and do not go outside these boundaries unless They are authorized to do so. The authentication techniques are based on the public key encryption method.

5) **Architecture Neutral**

Java was designed to support applications on heterogeneous network environments composed of a variety of processors, the operating system architectures, and multiple programming language interfaces. To enable a Java application to execute anywhere on the network, the Java compiler generates the bytecode instructions which are not dependent upon a particular computer architecture. These instruction are then interpreted on any machine and translated into the native machine code on the fly by the Java runtime.

6) **Portable**

Besides being architecturally neutral, Java is strict in its definition of the basic language. Unlike C or C++, there are no “implementation dependent” aspect of the specification. The size of the primitive data types are specified as is the behaviour of the arithmetic on them. For example, “int” always means a 32 bit integer. Thus, java programs are the same on any platform. There are no data type incompatibilities across the hardware and software architectures.

7) **Interpreted**

Java bytecode is not directly executed by the system, because Java is interpreted. However, the speed is more than adequate for most interactive application. Sun has recently unveiled a Java Chip that comes with a hard-coded Java interpreter and part of the runtime environment. Therefore, the speed will continue to improve further.

8) **Multithreaded**

Java was designed to meet the real world of creative, interactive, networked programs. Java’s multithreading capability provides the means to build applications with many concurrent threads of activity. The multithreading feature of Java has various sophisticated synchronization primitives. Moreover, Java’s high level system libraries have been written to be thread safe, i.e., the functionality provided by the libraries is available without conflict to multiple concurrent threads of execution.

9) **Dynamic**

While the Java compiler is strict in its static checking during compile time, the language and run time systems are dynamic during linking and loading stages. Classes are linked only as needed. New code modules can be linked in on demand from a variety of sources, even from sources across the network in a large number.

By making these interconnections between modules later, libraries can freely add new methods and instance variables without any effect on their client.

Check Your Progress 1

- 1) List the important Java buzzwords.

.....

.....

.....

- 2) What do you understand by Multithreading?

.....

.....

.....

- 3) How does Java programming language support robustness?

.....

.....

.....

1.4 THE JAVA PLATFORM

In general, a platform is a hardware or software environment in which a program runs. But here we are talking about it in context of software.

The Java platform has two components:

- the Java Virtual Machine (JVM)
- the Java Application Programming Interface (Java API).

The JVM is an abstract machine, designed to be implemented on top of the existing processors. The porting interface and adapters enable it to be easily ported to new operating systems without being completely overwritten. The bytecodes into which a Java Program is compiled run in this machine (JVM), and thus it is portable and architecture independent.

The Java API is a large collection of readymade software components that provide many useful features, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries (packages) of related components.

As a platform independent environment, Java can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time bytecode compilers can bring Java's performance code to that of native code without threatening portability.

1.5 JAVA LIBRARIES

The complete Java system includes a number of libraries of utility classes and methods of use to developers in creating multi-platform applications. These libraries are:

Java.lang The collection of base types (language type) that are always imported into any given compilation unit. This is where you will find the declarations of Object (the

root numbers of the class hierarchy) and Class, plus threads exceptions, wrappers for the primitive data types, and a variety of other fundamental classes.

Java.io: It is a collection of streams and random-access files. This is where you find the rough equivalent of the Standard I/O Library that you are familiar with on most UNIX systems. A further created library is called java.net, and it provides support for sockets, Telnet interfaces, and URLs.

Java.util: Container and utility classes. Here you will find classes such as Dictionary, Hash Table, and Stack, among others, plus encoder and decoder techniques, and date and time classes.

Java.awt: An Abstract Windowing Toolkit that provides an abstract layer enabling you to port java applications easily from one window system to another. This library contains classes for basic interface components such as events, colors fonts and controls such as buttons and scrollbars.

Java.net: This package provide classes for network programming. The system requires TCP/IP connection to work with java.net.



Check Your Progress 2

- 1) What is the basic purpose of Java.net package?

.....

- 2) List the important Java libraries.

.....

1.6 STARTING WITH JAVA

Let us consider the small java program given below and try to understand its various statements. We will also see how to compile and execute this program.

1. `//This is My First Program`
2. `public class My First Program{`
3. `public static void main (String args[]){`
4. `System.out.println ("This is my first program");`
5. `}`
6. `}`

Explanation

The first line is a single line comments, which will be ignored by the compiler. Comments are used in programs for documentation purpose. For multi-line comment, the comments should be enclosed between `/*` and `*/`.

In java everything should be in a class which is declared by the keyword **class**. Public is a keyword which makes this class available and accessible by any one, which means it can be accessed from any package, or any class.

My First Program is the name of the public class. This program should be saved in a file with a java extension. The primary name of the file should be the same as the name of the class, MyFirstProgram.java. Also note, that the case of the file name should be same as in the class name.

In the third line, a **static** keyword is used which signifies that this method will be called without creating an object of MyFirstProgram and this method will be invoked first and only once by the JVM. **void** is the return type of the main method, which means that this method is not going to return any value. The parameter that the main method takes is an array of string, which is used to take command line arguments.

In the fourth line, **System.out.println** method is used to print anything on the console.

Pair of { } braces defines a scope of any method or class.

In the Java programming language, a statement is a single line of code terminated with the semicolon (;)

To compile and run the program, type following instruction at the command prompt:

C:\>javacMyFirstProgram.java

C:\>javaMyFirstProgram

Result:

This is my first program

Javac is used to compile the program and **java** to run the program.

The necessary requirements for making a running program are:

- 1) The path of the jdk1.2\bin should be present.
- 2) The Name of the file in which the program is stored should be the same as the name in which the main method is present (with.java extension).



Check your Progress 3

- 1) Java is object oriented Programming language. What are its advantages?

.....
.....
.....
.....

- 2) What is the Java virtual machine (JVM)? Explain.

.....
.....
.....
.....

1.7 SUMMARY

In this introductory unit, you learned about the difference between java applet and java application, java buzzwords, and java libraries, etc.

Java programs can run from a web browser. Java programs that run from a web page are called java applets. Applets are modern GUIs, including buttons, text fields, text area menus, etc. Applets can respond to events such as mouse movements.

Unlike other conventional programming language, Java supports many features such as multithreaded, distributed, objected oriented, and robust programming language.

1.8 SOLUTIONS / ANSWERS

Check Your Progress 1

- 1) Java provides many interesting object oriented features which are also known as Java buzzwords. These features are:
 - (i) Java is simple and easy to learn
 - (ii) Java is object oriented programming language
 - (iii) Java is robust
 - (iv) Java is highly secure programming language
 - (v) Java is architectural neutral
 - (vi) Java is portable
 - (vii) Java is interpreted
 - (viii) Java is multithreaded
 - (ix) Java is dynamic.
- 2) Whenever a programme is run, the operating system creates a process. A thread is similar to a process concept, but it is a lightweight process. In a thread switching over from one task to another is much faster. Process however, is a heavyweight process. It takes more time in switching over from one task to another. The multithreading feature in Java allows writing the concurrent and distributed applications.
- 3) Java provides checking of the programme at different levels. It provides extensive compile-time checking followed by a second level of run-time checking. It also supports the exception handling feature in object-oriented style.

Check Your Progress 2

- 1) The basic purpose of the Java.net package is to write the network applications which support both TCP and UDP connections. For example one can simulate Telnet, FTP, and HTTP types of protocols.
- 2) Some of the important Java Libraries are:
 - (i) Java.lang
 - (ii) Java.io
 - (iii) Java.util
 - (iv) Java.awt
 - (v) Java.net

Check Your Progress 3

- 1) Object oriented programming language provides three main advantages:
 - i) Flexibility
 - ii) Modularity
 - iii) Reusability.

Using object oriented programming methodology, a program need not be redesigned from scratch. Using the reusability concept, S/W can be developed based on already existing object.

- 2) The Java virtual machine is a specification for an abstract computer. JVM consists of a class loader and a java interpreter that executes the bytecodes. The class loader loads class files from both the java program and java API for execution by the java interpreter. The Java interpreter may be a S/W interpreter that interprets the bytecodes one at a time or, it may be a just-in-time compiler that turns the architectural neutral bytecodes into native machine language for the host computer. The Java interpreter may be implemented in H/W.