

---

# UNIT 1: INTRODUCTION TO JAVA

---

## Unit Structure

- 1.0 Learning Objectives
- 1.1 Introduction
- 1.2 The Creation of Java
- 1.3 The Java Technology
- 1.4 Features of Java
- 1.5 Comparison of Java with C++
- 1.6 Garbage Collection
- 1.7 Creating a Java Program
- 1.8 Let Us Sum Up
- 1.9 Suggested Answer for Check Your Progress
- 1.10 Glossary
- 1.11 Assignment
- 1.12 Activities
- 1.13 Case Study
- 1.14 Further Readings

---

## 1.0 Learning Objectives

---

**After learning this unit, you will be able to understand:**

- Creation of Java
- Discuss Java Technology, Java Programming language and Java platform
- Define important terms of Java
- Compare Java with C++
- Describe garbage collection
- Illustrate Java program and provide comments

---

## 1.1 Introduction

---

The current programming problems are complex as beyond a certain size, structured programming cannot manage complexity. Using the concept of object oriented programming, complex programs can be organized using classes, inheritance and polymorphism. C++ was one of the popular programming languages for Object Oriented Programming.

With the advent of internet, Java was widely used. Initially, it was thought to develop a platform-independent language for consumer electronics like washing machines etc. Java became popular because the Web required platform-independent portable programs. In this unit, we'll discuss about the basics of Java Language.

---

## 1.2 The Creation of Java

---

### History of Java

James Gosling initiated the Java language project in June 1991 for use in one of his many set-top box projects. The language, initially called Oak after an oak tree that stood outside Gosling's office, also went by the name Green and ended up later renamed as Java, from a list of random words.

Sun released the first public implementation as Java 1.0 in 1995. It promised Write Once, Run Anywhere (WORA), providing no-cost run-times on popular platforms.

On 13 November 2006, Sun released much of Java as free and open source software under the terms of the GNU General Public License (GPL).

On 8 May 2007, Sun finished the process, making all of Java's core code free and open-source, aside from a small portion of code to which Sun did not hold the copyright.



**Java logo**

## Check your progress 1

1. Who initiated Java language?
2. What do the terms WORA and GPL mean?

.....

.....

.....

.....

.....

.....

.....

.....

.....

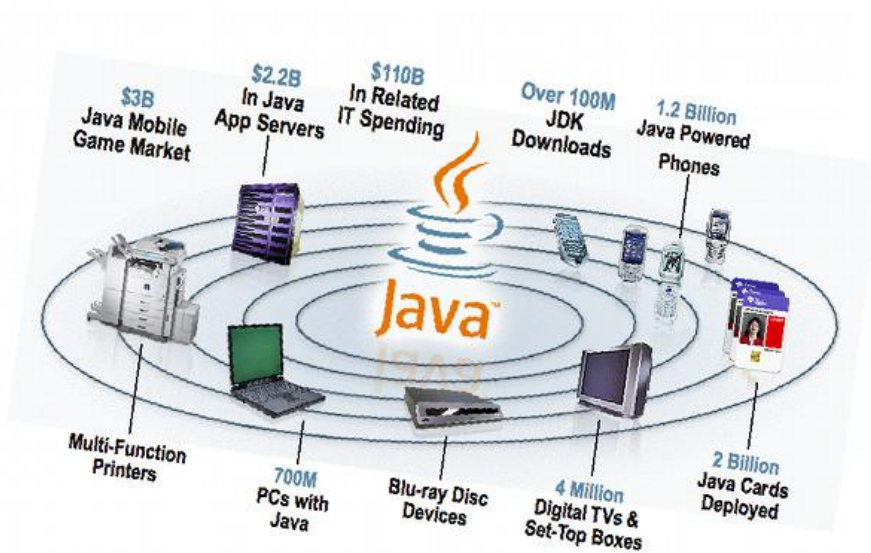
.....

.....

---

## 1.3 The Java Technology

---



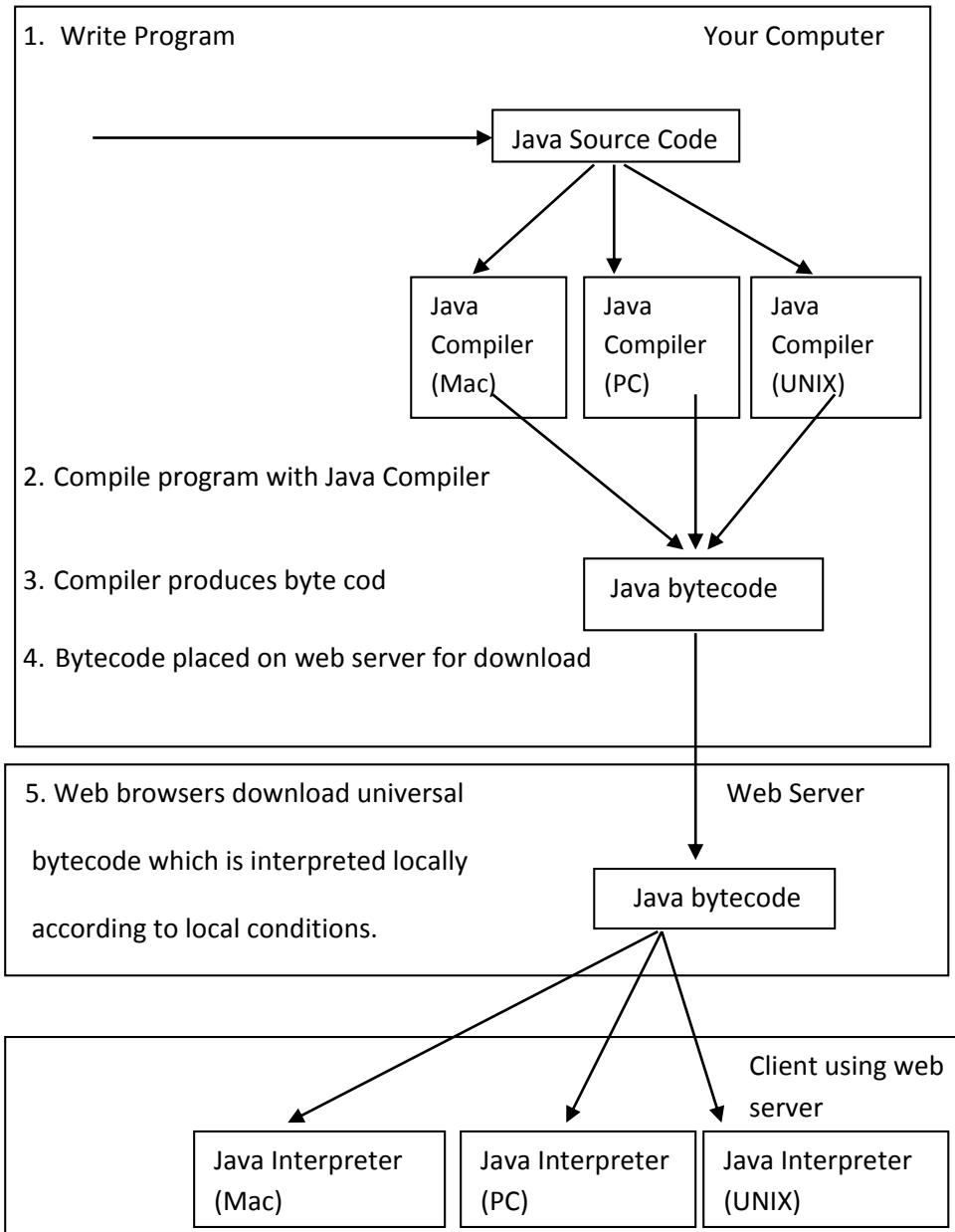
The Java Technology

The Java technology is a programming language and a platform. Let us discuss Java as a programming language and a platform both.

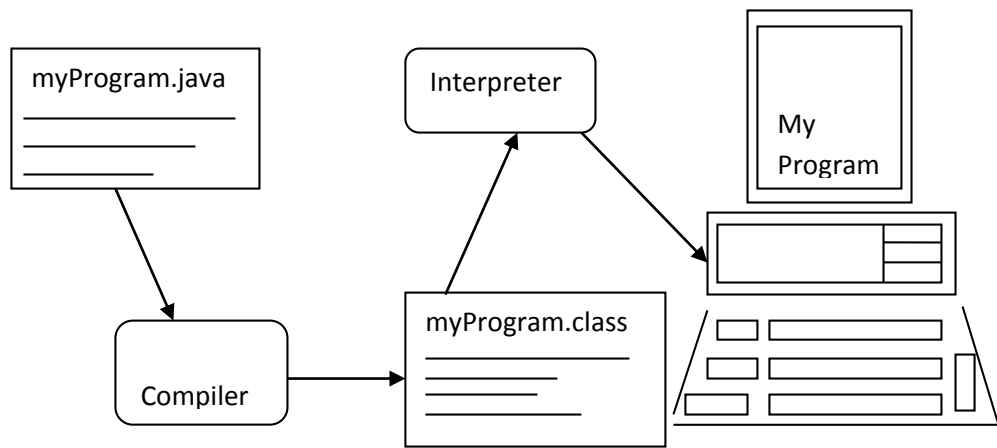
### **Java Programming Language**

Java is simple, object-oriented, multi-threaded, robust, portable and dynamic programming language. It is an object-oriented language similar to C++ but simplified to eliminate language features which cause common programming errors.

The Java source code files (files with .java extension) are compiled into a format called byte code (files with a .class extension) which can then be executed by a Java interpreter. The compiled Java code can run on most computers because Java interpreters and runtime environments which are known as Java Virtual Machines exist for most operating systems including UNIX, Macintosh OS and Windows.



**Java Programming Environment**



**Execution of Java Program**

### **The Java Platform**

A platform can be defined as the hardware or software environment in which a program runs. Most of the platforms can be described as a combination of the operating system and hardware. The Java platform differs from most of the other platforms. It is a software only platform which runs on the top of other hardware based platforms. The Java platform has two components:

The Java Virtual Machine (JVM)

The Java Application Programming Interface (Java API)

The JVM is the base for the Java platform and is ported onto various hardware based platforms.

The Java API is a large collection of ready-made software components which provide many useful capabilities, such as Graphical User Interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages

### Check your progress 2

1. What do you mean by platform?
2. Name the two components of Java platform.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

---

## 1.4 Features of Java

---

Although the main features of Java are its portability and security. Apart from this, certain other features are also there which are discussed below:

1. **Simple** - Java is primarily derivative of C++. It omits rarely used and confusing features of C++ such as preprocessor, operator overloading, multiple inheritance etc.
2. **Object Oriented** - Java is an object oriented programming language, in which data is treated as objects to which methods are applied. Its basic execution unit is the class.

3. **Robust** - Java is strongly typed language. The type checking is carried out at both compile and runtime with every data structure clearly defined and typed. Java supports automatic garbage collection which manages memory by preventing memory leaks.
4. **Architecture Independent** - The Java compiler compiles source code and generates bytecode which is intermediate between source and machine code. These machine codes are neutral and have nothing to do with particular computer architecture. The Java Virtual Machine (JVM) converts the bytecode into native code for a particular processor.
5. **Portable** - The interpreter for the Java Virtual Machine can be ported to any computer hardware/operating system, so that all the codes compiled for it will run on that system. This forms the basis for Java's portability.
6. **Multithreaded** - Multithreading helps to overcome the performance problems caused due to interpreted code as compared to the main code. Since an executing program hardly ever uses CPU cycles 100 percent of the time, Java uses the idle time to perform the necessary garbage cleanup and general system maintenance that renders.
7. **High Performance** - With the use of Just-In-Time compilers Java enables high performance.
8. **Distributed** - Java is designed for the distributed environment of the internet.
9. **Dynamic** - Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.
10. **Secure** - With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.



**Check your progress 3**

1. Explain the multi-threaded and dynamic feature of Java.
2. Write a note on portability and security feature of Java.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

---

**1.5 Comparison of Java with C++**

---

Java is somewhat similar to C++. However, Java is not a superset or subset of C++; it can be seen as a derivation with many modifications and extensions. The given table displays the difference between them:

Table 1.1: C++ Vs. Java

C++	Java
Hybrid between procedural and object-oriented language	Purely object-oriented language
No Automatic garbage collection	Automatic garbage collection
Programs are compiled	Programs are compiled and interpreted
Architecture specific	Architecture neutral

Supports multiple inheritance	No multiple inheritance
Supports operator Overloading	Does not support operator overloading
Templates as parameterized type	No parameterized type
Supports Pointers with dereferencing (* or ->) and address (&) operators.	No explicit pointer manipulation and no pointer arithmetic
Strings are null-terminated character arrays	Strings are objects
Main function can return a value	Main method cannot return a value

**Check your progress 4**

1. Compare Java with C++ with respect to garbage collection and inheritance.
2. Give the difference between Java and C++ with respect to strings and operator overloading.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



**Garbage Collection**

In C++ language, the dynamically allocated objects are manually released by the use of a delete operator. In Java, the deallocation process is automatically done and the technique to do that is called garbage collection.

When no references to an object exist, that object is no longer needed and the memory occupied by the object can be reclaimed. The explicit destruction of objects is not done as in C++.

Garbage collection occurs at irregular intervals during the execution of program. It will not occur simply because one or more objects exist which are no longer needed.

Different Java run-time implementations take varying approaches to garbage collection, so it is recommended that one should not think about it while writing programs.

### **Check your progress 5**

1. Explain the deallocation process in Java and C++.
2. What is the occurrence of garbage collection?

.....

.....

.....

.....

---

## 1.7 Creating a Java Program

---

The execution of Java Program is divided into several steps. These steps are discussed below:

- a. Writing the Code: The steps performed while writing the code are:
  1. Use any text editor
  2. Save the program as.java
  3. Name the program the same as the class containing main

Compiling the Code: The process of compiling the code involves the given steps:

- Use the Command prompt
- Java c programname.java
- Creates programname. class file

b. Interpreting the ByteCodes: The ByteCodes are interpreted by following the given steps:

- Use the command prompt
- .java programname

After studying about the steps of creating a Java Program, let us write a small and simple program to display “I like Studying Java” on the screen.

### **// Program to display message on screen**

```
class display
{
public static void main (String [ ] args)
{
System.out.println (“I like Studying Java”);
//Displays the string
}
}
```

Now let us try to understand the above program. The first point of execution of any Java program is its main method. This main method has to be defined within a class.

In order to run an application with the Java interpreter, the name of the class has to be specified which has to be executed. The interpreter invokes the main method defined within that class.

The method signature for the main method contains three modifiers:

- **Public** indicates that the main method can be accessed outside the class in which it is declared.
- **Static** indicates that the main method can be invoked without creating an instance of the class.
- **Void** indicates that the main method doesn't return any value.

The main method accepts a single argument, that is, an array of elements of class `String` and `arg` receives command line arguments. The next line is `System.out.println ("I like Studying Java")`, this statement displays the string "I Like Studying Java".

The output gets displayed by `println ( )` method. The `println ( )` displays the string which is passed to it. This method can also be used to display other types of information also. It begins with `System.out`; here `System` is class and `out` is the output stream which is connected to the console. Thus, `System.out` is an object which encapsulates console output.

## Providing Comments

Comments are used to provide brief documentation to the program. There are two most commonly used methods of providing comments to Java program:

- a. **Single Line Comment** - The single line comment is used to provide comments of one line. It starts with `//` and ends with new line character. For example,

```
// this is my first Java Program
```

- b. **Multiline Comment** - When the comment to be included is of multiple lines, then multiline comment is used. It starts with `/*` and ends with `*/`. For example,

```
/* This is  
My first  
Java Program */
```

**Check your progress 6**

1. Explain the steps involved in the execution of Java program.

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

---

**1.8 Let Us Sum Up**

---

This Unit gives lot of basic learning right from History of Java that James Gosling initiated the Java language project in June 1991 for use in one of his many set-top box projects. The language, initially called Oak after an oak tree that stood outside Gosling's office, also went by the name Green and ended up later renamed as Java.

As the current programming problems are complex, as beyond a certain size, structured programming cannot manage complexity. Using the concept of object oriented programming, complex programs can be organized using classes, inheritance and polymorphism. C++ was one of the popular programming languages for Object Oriented Programming.

Java is simple, object-oriented, multi-threaded, robust, portable and dynamic programming language. It is an object-oriented language similar to C++ but simplified to eliminate language features which cause common programming errors.

A platform can be defined as the hardware or software environment in which a program runs. Most of the platforms can be described as a combination of the operating system and hardware. The Java platform has two components:

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

Although the main features of Java are its portability and security. Apart from this, certain other features are also there which are discussed below:

1. Simple and
2. Object Oriented

About Garbage collection in C++ language, the dynamically allocated objects are manually released by the use of a delete operator. In Java, the deallocation process is automatically done and the technique to do that is called garbage collection.

Creating a Java Program is the execution of Java Program is divided into several steps. These steps are discussed below:

- a. Writing the Code: The steps performed while writing the code are:
  - Use any text editor
  - Save the program as .java
  - Name the program the same as the class containing main
- b. Compiling the Code: The process of compiling the code involves the given steps-
  - Use the Command prompt
  - `.javac programname.java`
  - Creates `programname.class` file

- c. Interpreting the ByteCodes: The ByteCodes are interpreted by following the given steps:
- Use the command prompt
  - `.java programname`

Comments are used to provide brief documentation to the program. There are two most commonly used methods of providing comments to Java program

**Single Line Comment** The single line comment is used to provide comments of one line. It starts with `//` and ends with new line character. **Multiline Comment:** When the comment to be included is of multiple lines, then multiline comment is used. It starts with `/*` and ends with `*/`.

---

## 1.9 Suggested Answer for Check Your Progress

---

**Check your progress 1**

**Answers: See Section 1.2**

**Check your progress 2**

**Answers: See Section 1.3**

**Check your progress 3**

**Answers: See Section 1.4**

**Check your progress 4**

**Answers: See Section 1.5**

**Check your progress 5**

**Answers: See Section 1.6**

**Check your progress 6**

**Answers: See Section 1.7**



---

## 1.10 Glossary

---

1. **Java Virtual Machines** - The compiled Java code can run on most computers because Java interpreters and runtime environments which are known as Java Virtual Machines
2. **Object Oriented** - Data is treated as objects to which methods are applied. Its basic execution unit is the class.
3. **Portable** - The interpreter for the Java Virtual Machine can be ported to any computer hardware/operating system, so that all the codes compiled for it will run on that system. This forms the basis for Java's portability.
4. **Dynamic** - Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and
5. **Java Platform** - The hardware or software environment in which a program runs.

---

## 1.11 Assignment

---

Discuss the function of modifiers in the method signature of main method.

---

## 1.12 Activities

---

Find the new terminologies and write down the meaning and importance of each.

---

## 1.13 Case Study

---

Write a program to display a message on screen and explain the execution of the program

---

## 1.14 Further Reading

---

1. Core Java 2, 2 volumes, Cay S. Horstmann, Gary Cornell, The Sun Microsystems Press, 1999, Indian reprint 2000
2. Java 2, The Complete Reference, Patrick Naughton and Herbert Schildt, Tata McGraw Hill, 1999
3. Programming with Java, Ed. 2, E. Balagurusamy, Tata McGraw Hill, 1998, reprint, 2000
4. The Java Tutorial, Ed. 2, 2 volumes, Mary Campione and Kathy Walrath, Addison Wesley Longmans, 1998
5. The Java Language Specification, Ed. 2, James Gosling, Bill Joy, Guy Steele & Gilad Bracha, (Ed. 1 1996, Ed. 2 2000), Sun Microsystems, 2000
6. Using Java 2, Joseph L. Weber, Prentice Hall, Eastern Economy Edition, 2000