

Unit-1

Overview of Java Language

Java is a general-purpose, object-oriented programming language. We can develop two types of java programs :

- a) Stand-alone applications
- b) Web applets

Stand-alone applications are programs written in java to carry out certain tasks on a stand-alone computer. In fact, Java can be used to develop programs for all kinds of applications. Executing a stand-alone Java program involves two steps

1. Compiling source code into bytecode using javac compiler.
2. Executing the bytecode program using java interpreter.

Source code ----- byte code ----- machine code
 javac java

Simple Java program

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

1) class is a keyword and declares that a new class definition follows. Simple is a Java identifier that specifies the name of the class to be defined.

2) Every class definition in Java begins with an opening brace “{” and ends with a matching closing brace “}”, appearing in the last line.

3) Every java application program must include the main() method. This is the starting point for the interpreter to begin the execution of the program. A Java application can have any number of classes but only one of them must include a main method to initiate the execution.

Note :- Java applets will not use the main method at all.

public -> The keyword public is an access specifier that declares the main method as unprotected and therefore making it accessible to all other classes.

static -> It declares this method as one that belongs to the entire class and not a part of any objects of the class. The main must always be declared as static since the interpreter uses this method before any objects are created.

void -> The Type modifier void states that the main method does not return any value.

String args[] -> It declares a parameter named args, which contains an array of objects of the class type String.

System.out.println() -> The println method is a member of the out object, which is a static data member of System class. The method println always appends a newline character to the end of the string. This means that any subsequent output will start on a new line. Every Java statements must end with a semicolon.

Comments

Java permits both the single-line comments and multi-line comments. The single-line comments begin with // and end at the end of the line. For multi-line comments starting with a /* and ending with a */ .

Java Program Structure

Documentation Section	-> Suggested
Package Statement	-> Optional
Import Statements	-> Optional
Interface Statements	-> Optional
Class Definitions	-> Optional
main method class	
{	-> Essential
main method definition	
}	

Documentation Section

The documentation section comprises a set of comments lines the name of the program, the author and other details, which the programmer would like to refer to at a later stage.

Package Statement

The first statement allowed in a java file is a package statement. This statement declares a package name and informs the compiler that the classes defined here belong to this package. For example,

```
package student;
```

Import Statements

It is used to import a package and all its classes.

Interface Statements

An interface is like a class but includes a group of method declarations. This is also an optional section and is used only when we wish to implement the multiple inheritance feature in the program.

Class Definitions

A Java program may contain multiple class definitions. Classes are the primary and essential elements of a Java program. These classes are used to map the objects of real-world problems.

Main Method Class

Since every Java stand-alone program requires a main method as its starting point, this class is the essential part of a java program. A simple Java program may contain only this part. The main method creates objects of various classes and establishes communications between them. On reaching the end of main, the program terminates and the control passes back to the os.

WAP to add two numbers.

```
class add
{
    public static void main(String args[])
    {
        int a=5;
        int b=10,c;
        c=a+b;
        System.out.println("Add="+c);
    }
}
```

Use of Math functions

```
import java.lang.Math;
```

The purpose of this statement is to instruct the interpreter to load the Math class from the package lang.

WAP to calculate the square root of number.

```
import java.lang.Math;
class sqroot
{
    public static void main(String args[])
    {
        double x=5;
        double y;
        y = Math.sqrt(x);
        System.out.println("y=" + y);
    }
}
```

WAP to Calculate the Power of any number.

```
Import java.lang.Math;
class power
{
    public static void main(String args[])
    {
```

```

int num=5;
int b;
b=Math.pow(num,4);
System.out.println("Power of number=" +b);
}
}

```

Write a Java program to swap two numbers .

a) with taking third variable

```

class swap
{
    public static void main(String args[])
    {
        int a,b,c;
        a=10;
        b=12;
        c=a;
        a=b;
        b=c;
        System.out.println("After swapping");
        System.out.println("a=" +a);
        System.out.println("b=" +b);
    }
}

```

b) without taking third variable

```

class swap
{
    public static void main(String args[])
    {
        int a,b,c;
        a=10;
        b=12;
        a=a+b;
        b=a-b;
        a=a-b;
        System.out.println("After swapping");
        System.out.println("a=" +a);
        System.out.println("b=" +b);
    }
}

```

Java Tokens

Smallest individual units in a program are known as tokens. The compiler recognizes them for building up expressions and statements.

In Simple terms, a Java program is a collection of tokens, comments and white spaces. Java language includes five types of tokens. They are:

- Reserved Keywords
- Identifiers
- Literals
- Operators
- Seperators

Java Character Set

Java characters are defined by the Unicode Character Set. The Unicode is a 16-bit character coding system.

Keywords

Keywords are identifiers, such as public, static and class that have a special meaning inside Java source code, and outside of comments and Strings. Keywords are reserved for their intended use and cannot be used by the programmer for variable, or method names. Some of the reserved keywords in

boolean	continue	default	do	double	else	extends	final
	finally	break	byte	case	for	goto	if
	implements	import	interface				
native	new	null	package	protected	public	return	
	instance of	int	catch	char	short		
static	super	switch	synchronized	this	throw		
throws	class	transient	try	void	volatile	while	
	abstract						

Identifiers in Java

Identifiers are the names of **variables, methods, classes, packages** and **interfaces**. Identifiers must be composed of **letters, numbers**, the **underscore_** and the **dollar sign \$**. They cannot contain white spaces. Identifiers may only begin with a letter, the underscore or a dollar sign. we cannot begin a variable name with a number. All variable names are case sensitive.

for example:

My **Variable** is not the same as my **Variable**.
There is no limit to the length of a Java variable name.

Identifiers must be meaningful, short enough to be quickly and easily typed and long enough to be descriptive and easily read. Java developers have followed some naming conventions.

1. Names of all public methods and instance variables start with a leading lowercase letter.
Ex- average, sum

2. When more than one words are used in a name, the second and subsequent words are marked with a leading uppercase letters.

Ex : dayTemperature
totalMarks

3. All private and local variables use only lowercase letters combined with underscores.

Ex : length
batch_strength

4. All classes and interfaces start with a leading uppercase letter (and each subsequent word with a leading uppercase letter).

Ex : Student
HelloJava
Vehicle

5. Variables that represent constant values use all uppercase letters and underscores between words.

Ex : TOTAL
F_MAX