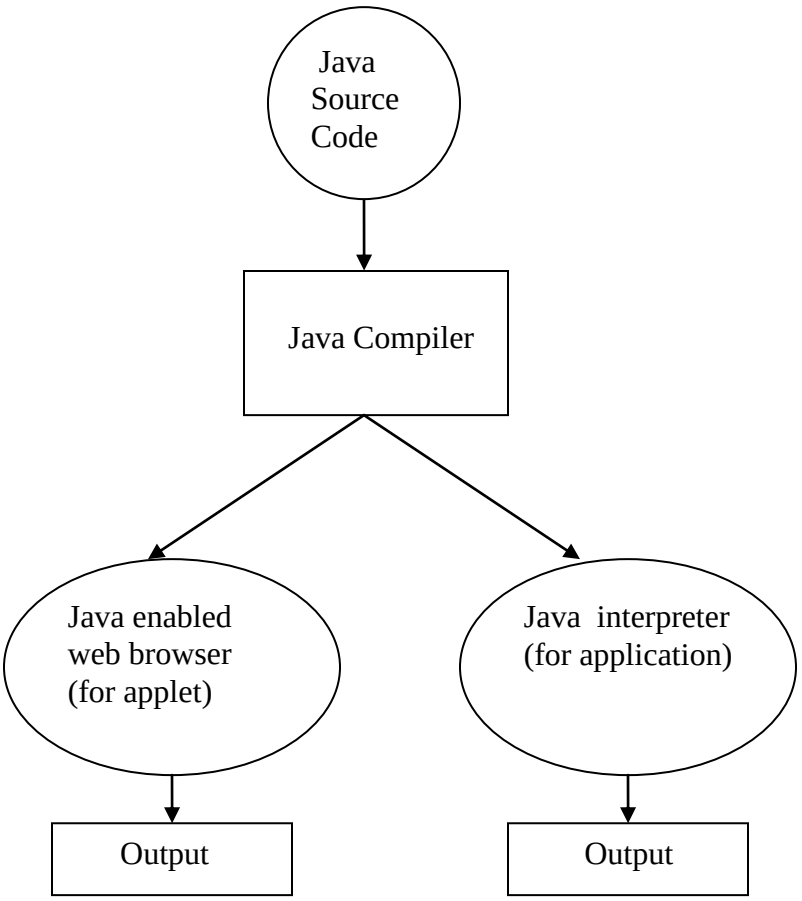
Overview of Java Language

Java is a general-purpose, object-oriented programming language. We can develop two types of java programs- 1) Stand-alone applications, 2) Web applets.



Stand-alone applications are programs written in Java to carry out certain tasks on a stand-alone local computer. Executing a stand-alone Java program involves two stapes-
1. Compiling source code into bytecode using **javac** compiler.
2. Executing the bytecode program using **java** interpreter.

Now we will consider some simple application programs, which could demonstrate the general structure of Java application programs. We will also discuss here the basic elements of Java language and steps involved in executing a Java application program.

- First Java program: Consider the first Java program-

```
class Firstprogram
{
  Public static void main(String args[])
  {
    System.out.println("Welcome to the world of Java!");
   }
 }
```

- Explanation:

1. The first line *class Firstprogram* declares a class, which is an object-oriented construct. Since Java is a true object oriented language, everything must be placed inside a class. Class is a keyword and declares that a new class definition follows. *Firstprogram* 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 in the example.

3. The third line *public static void main(String args[])* defines a method named main. 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. The line also contains a number of keywords, public, static, void.

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

Next appears the keyword *static* which declares this method as one that belongs to the entire class and not a part of any objects of the class. The main() always be declared as static since the interpreter uses this method before any objects are created.

The modifier void states that the main() method does not return any value.

- **Java tokens:** A Java program is basically a collection of classes. A class is defined by a set of declaration statements and methods containing executable statements. Most statements contain expressions which describe the action carried out on data. *Smallest individual units in a program are known as tokens.* The compiler recognizes them for building up expression and statements.
  Java language includes five types of tokens. They are-
  - Reserved keywords
  - Identifiers
  - Literals
  - Operators
  - Separators

- **Java character set:** The smallest units of java language are the characters used to write Java tokens. These characters are defined by the Unicode character set, an emerging standard that tries to create characters for a large number of scripts worldwide.

  The Unicode is a 16 bit character coding system and currently supports more than 34,000 defined characters derived from 24 languages. However, most of us use only the basic ASCII characters (a subset of UNICODE character set) which include letters, digits and punctuation marks used in normal English.

- **The Java Keywords:** There are 49 reserved keywords currently defined in the Java language. These keywords, combined with the syntax of the operators and separators, form the definition of the Java language. These keywords cannot be used as names for a variable, class, or method.

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