SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

OBJECT ORIENTED PROGRAMMING USING JAVA

III YEAR - IV SEM

UNIT 1 – Fundamentals of Java

TOPIC 1 – Java Basics

Introduction

Test.java

```
1  package com.journaldev.java.examples1;
2
3  public class Test {
4
5      public static void main(String args[]) {
6
7          System.out.println("Hello World");
8
9      }
10 }
```

Introduction

- **public:** It is an access specifier. We should use a public keyword before the main() method so that JVM can identify the execution point of the program. If we use private, protected, and default before the main() method, it will not be visible to JVM.

- **static:** You can make a method static by using the keyword static. We should call the main() method without creating an object. Static methods are the method which invokes without creating the objects, so we do not need any object to call the main() method.

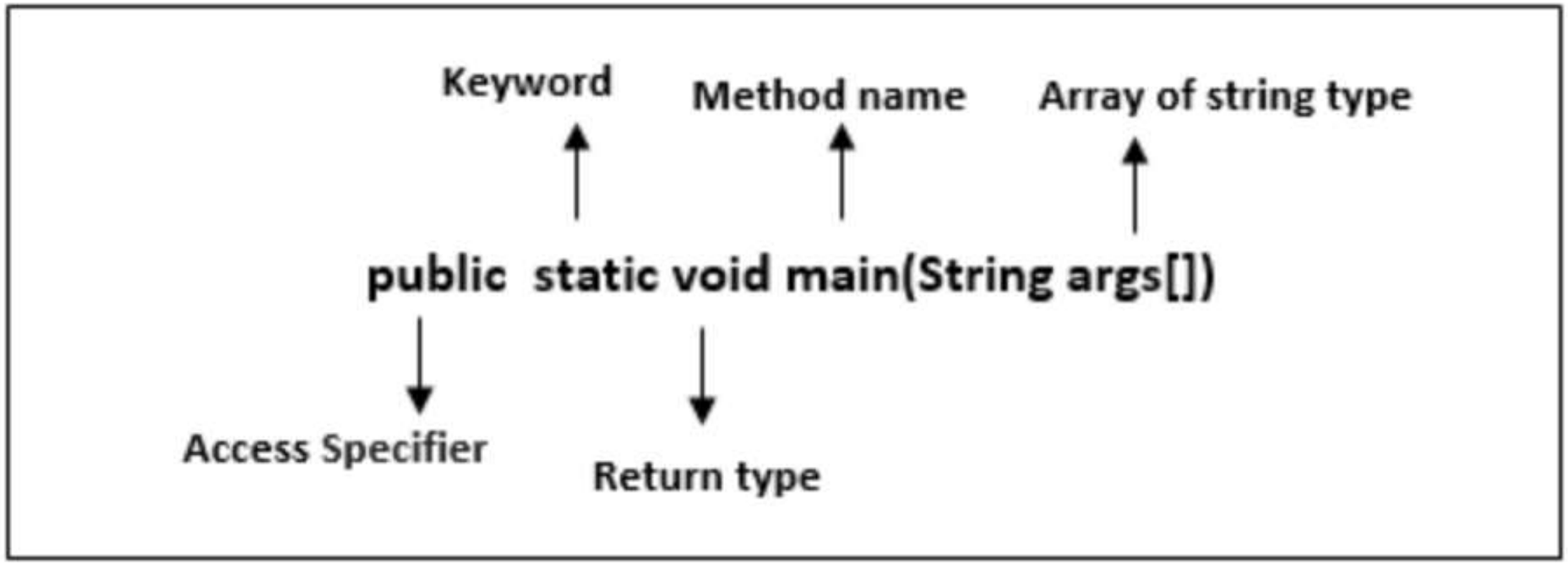
Introduction

- **void:** In Java, every method has the return type. Void keyword acknowledges the compiler that main() method does not return any value.

- **main():** It is a default signature which is predefined in the JVM. It is called by JVM to execute a program line by line and end the execution after completion of this method. We can also overload the main() method.

- **String args[]:** The main() method also accepts some data from the user. It accepts a group of strings, which is called a string array. It is used to hold the command line arguments in the form of string values.

Introduction



PUBLIC

```
public class Test {

static void main(String[] args){

        System.out.println("Hello World");

}
}
```

```
$ javac Test.java
$ java Test
Error: Main method not found in class Test, please define the main method as:
   public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
$
```

Static

```
public class Test {

public void main(String[] args){

        System.out.println("Hello World");

}
}
```

```
$ javac Test.java
$ java Test
Error: Main method is not static in class Test, please define the main method as:
   public static void main(String[] args)
$
```

Main

```
public class Test {

public static void mymain(String[] args){

        System.out.println("Hello World");
}
}
```

And we try to run this program, it will throw an error that the main method is not found.

```
$ javac Test.java
$ java Test
Error: Main method not found in class Test, please define the main method as:
   public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
$
```

Blocks

```
class  Demo
{
static                  //static block
{
System.out.println("Static block");
}
public static void main(String args[])  //static method
{
System.out.println("Static method");
}
}
```

Output:

```
Static block
Static method
```

Main

```
class DemoStaticBlock
{
Static                          //static block
{
System.out.println("Static block");
}
}
```

Output:

```
Error: Main method not found in the class Demo, please define the main method as:
public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
```

Concepts

```
class OverloadMain
{
public static void main(int a)  //overloaded main method
{
System.out.println(a);
}
public static void main(String args[])
{
System.out.println("main method incoked");
main(6);
}
}
```

Output:

```
main method invoked
6
```

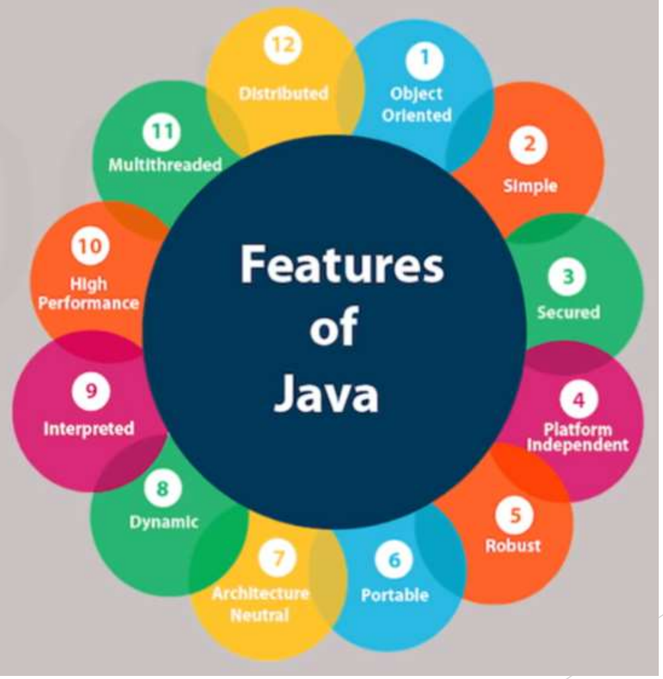
JAVA FEATURES

Features of Java

The features of Java are also known as java *buzzwords*.

A list of most important features of Java language is given below.

1) Simple
2) Object-Oriented
3) Portable
4) Platfom Independent
5) Secured
6) Robust
7) Architecture neutral
8) Interpreted
9) High Performance
10) Multithreaded
11) Distributed
12) Dynamic



Concepts

Simple

Java is very easy to learn, and its syntax is simple, clean and easy to understand. According to Sun, Java language is a simple programming language because:

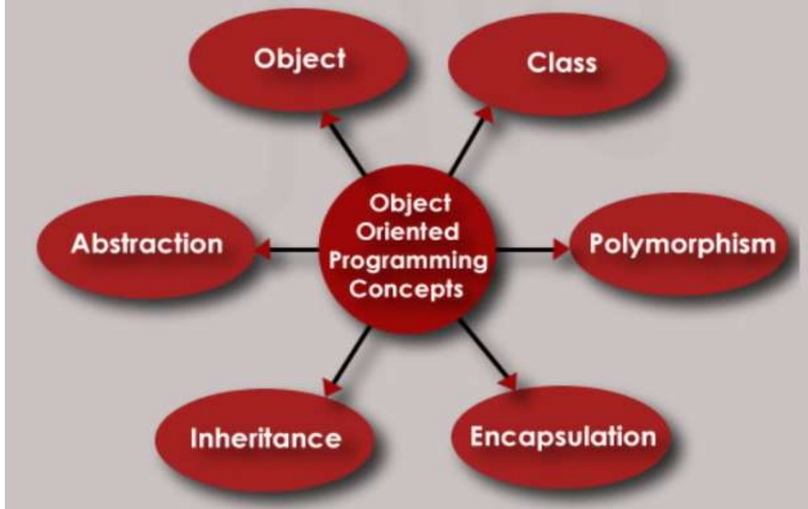
- Java syntax is based on C++ (so easier for programmers to learn it after C++).
- Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.
- There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

Concepts

Object-oriented

Java is an object-oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behavior.

Object-oriented programming (OOPs) is a methodology that simplifies softwaredevelopment and maintenance by providing some rules.



Basic concepts of OOPs are:

1) Object

2) Class

3) Inheritance

4) Polymorphism

5) Abstraction

6) Encapsulation

Concepts

Platform Independent

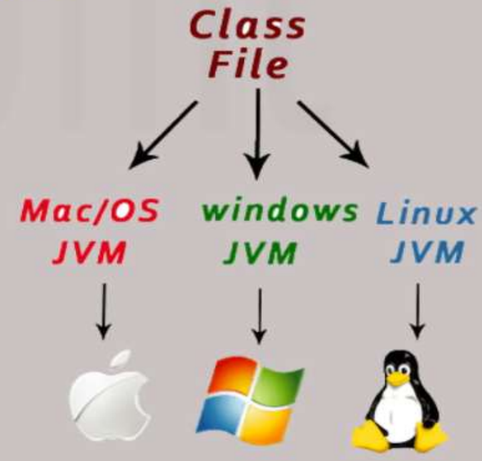
Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.

There are two types of platforms software-based and hardware-based. Java provides a software-based platform.

It has two components:

1) Runtime Environment
2) API(Application Programming Interface)

Java code can be run on multiple platforms, for example, Windows, Linux, Sun Solaris,Mac/OS, etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms, i.e., Write Once and Run Anywhere(WORA).



Concepts

Secured

Java is best known for its security. With Java, we can develop virus-free systems.

Java is secured because:

- No explicit pointer
- Java Programs run inside a virtual machine sandbox
- Classloader :

  Classloader in Java is a part of the Java Runtime Environment(JRE) which is used to load Java classes into the Java Virtual Machine dynamically. It adds security by separating the package for the classes of the local file system from those that are imported from network sources.

- Bytecode Verifier: It checks the code fragments for illegal code that can violate access right to objects.
- Security Manager: It determines what resources a class can access such as reading and writing to the local disk.

Concepts

Robust

Robust simply means strong. Java is robust because:

- It uses strong memory management.
- There is a lack of pointers that avoids security problems.
- There is automatic garbage collection in java which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.
- There are exception handling and the type checking mechanism in Java. All these points make Java robust.

Concepts

Architecture-neutral

Java is architecture neutral because there are no implementation dependent features, for example, the size of primitive types is fixed.

In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. However, it occupies 4 bytes of memory for both 32 and 64-bit architectures in Java.

Portable

Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.

Concepts

High-performance

Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code.

It is still a little bit slower than a compiled language (e.g., C++).
Java is an interpreted language that is why it is slower than compiled languages

e.g., C, C++, etc.

Distributed

Java is distributed because it facilitates users to create distributed applications in Java.

RMI and EJB are used for creating distributed applications.

This feature of Java makes us able to access files by calling the methods from any machine on the internet.

Concepts

Multi-threaded

A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads.

The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications, etc.

Dynamic

Java is a dynamic language. It supports dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.

Java supports dynamic compilation and automatic memory management (garbage collection).