



MEHRAN UNIVERSITY
OF ENGINEERING & TECHNOLOGY
JAMSHORO, PAKISTAN

Object Oriented Programming

Introduction to Java

Lecture#01

Java Basics

Lecture Slides by Engr. Mehwish Shaikh

What is Java?

- Java is a **programming language** and a **platform**.
- Java is a high level, robust, secured and object-oriented programming language.
- **Platform:** Any hardware or software environment in which a program runs, is known as a platform. Since Java has its own runtime environment (JRE) and API, it is called platform.

Introduction to Java programming language

- Today Java programming language is one of the most popular programming language which is used in critical applications like stock market trading system on BSE, banking systems or android mobile application.
- Java was developed by James Gosling from Sun Microsystems in 1995 as an object-oriented language for general-purpose business applications and for interactive, Web-based Internet applications. The goal was to provide platform-independent alternative to C++.

Introduction to Java programming language

- In other terms it is architecturally neutral, which means that you can use Java to write a program that will run on any platform or device (operating system).
- Java program can run on a wide variety of computers because it does not execute instructions on a computer directly. Instead, Java runs on a Java Virtual Machine (JVM).

Why Java?

- Java has been tested, refined, extended, and proven by a dedicated community of Java developers, architects and enthusiasts.
- **Simple Grammar** – Java has a very simple grammar familiar to anyone with experience in C and C++
- **Portability** – These days Java really does run well on all the popular platforms
- **Speed** – The latest JIT compilers for Sun's JVM approach the speed of C/C++ code, and in some memory allocation intensive circumstances, exceed it.
- **Garbage Collection** – the programmer doesn't have to worry about memory (most of the time)
- **Huge library** and developer community support available on Internet.

Where it is used?

- According to Sun, 3 billion devices run java. There are many devices where java is currently used. Some of them are as follows:
 - Desktop Applications such as acrobat reader, media player, antivirus etc.
 - Web Applications such as irctc.co.in, javatpoint.com etc.
 - Enterprise Applications such as banking applications.
 - Mobile
 - Embedded System
 - Smart Card
 - Robotics
 - Games
 - etc.

Overview of Java

- **Java divided into three categories, they are :**
 - J2SE (Java 2 Standard Edition)
 - J2EE (Java 2 Enterprise Edition)
 - J2ME (Java 2 Micro or Mobile Edition)

Overview of Java

- **J2SE**

- J2SE is used for developing client side applications.

- **J2EE**

- J2EE is used for developing server side applications.

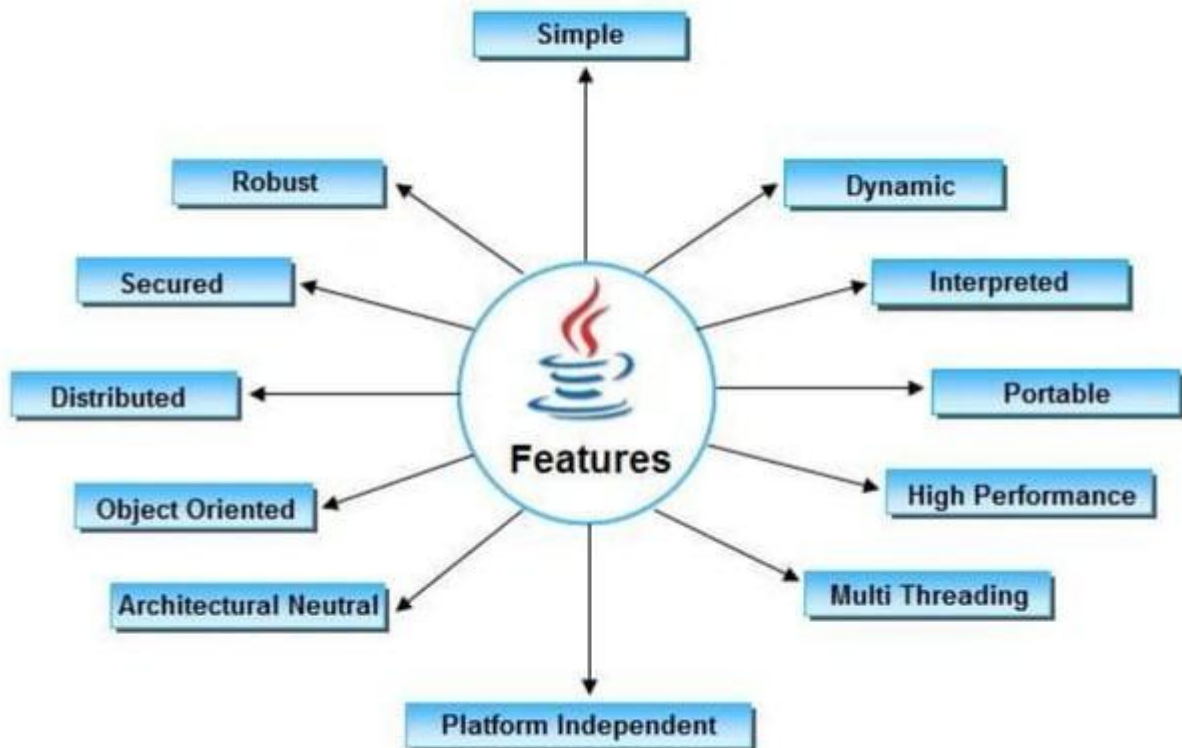
- **J2ME**

- J2ME is used for developing mobile or wireless application by making use of a predefined protocol called WAP(wireless Access / Application protocol).

Features of Java

- There is given many features of java. They are also known as java buzzwords. The Java Features given below are simple and easy to understand.
 - Simple
 - Object-Oriented
 - Platform independent
 - Secured
 - Robust
 - Architecture neutral
 - Portable
 - Dynamic
 - Interpreted
 - High Performance
 - Multithreaded
 - Distributed

Introduction to Java Programming Language



Features of Java

- **Simple**

- According to Sun, Java language is simple because:
 - Syntax is based on C++ (so easier for programmers to learn it after C++).
 - Removed many confusing and/or rarely-used features e.g., explicit pointers, operator overloading etc.
 - No need to remove unreferenced objects because there is Automatic Garbage Collection in java.
 - And also removed many other confusing and/or rarely used features like explicit pointer, operator overloading etc.

Features of Java

- **Object-oriented**

- 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 simplify software development and maintenance by providing some rules.
- Basic concepts of OOPs are:
 - Object
 - Class
 - Inheritance
 - Polymorphism
 - Abstraction
 - Encapsulation

Features of Java

- **Platform Independent**

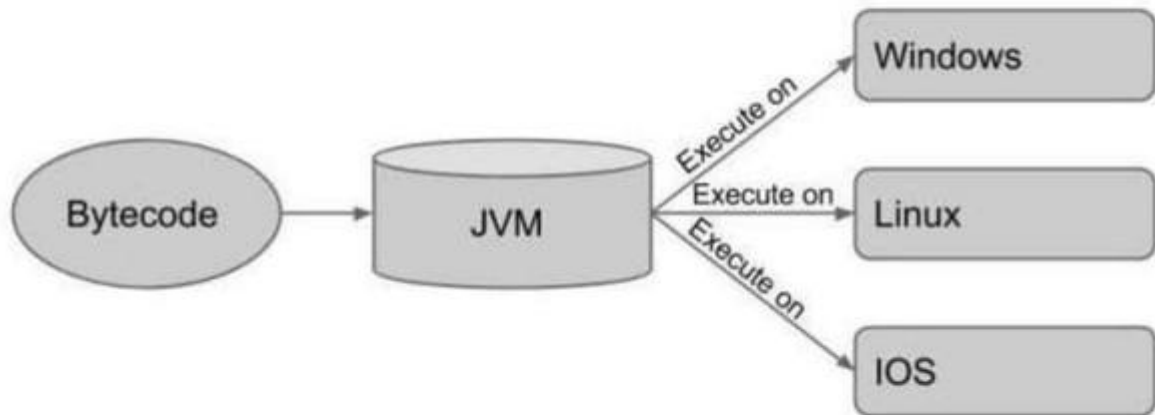
- A platform is the hardware or software environment in which a program runs.
- There are two types of platforms software-based and hardware-based. Java provides software-based platform.
- The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware-based platforms. It has two components:
 - Runtime Environment
 - API(Application Programming Interface)

Features of Java

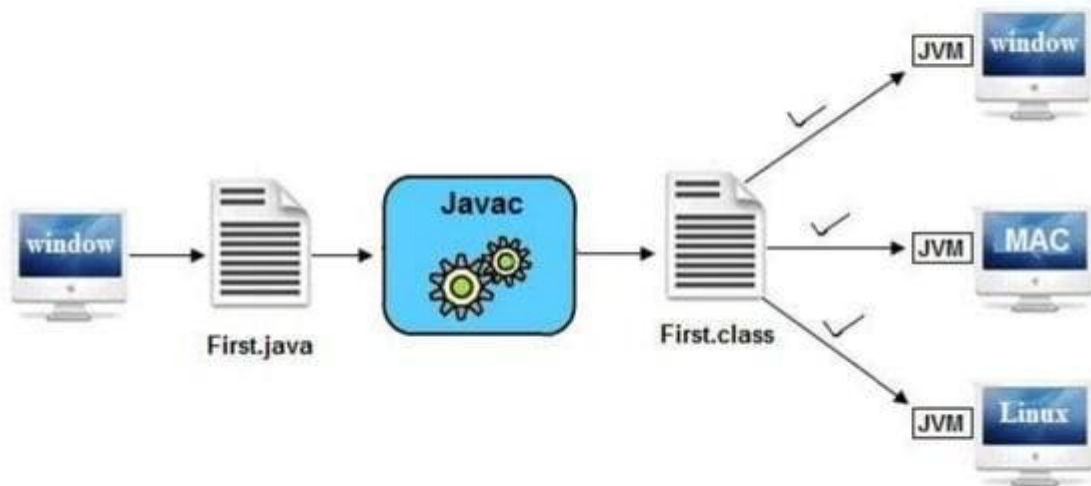
- **Platform Independent**

- Java code can be run on multiple platforms e.g. 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).
- Java solves the problem of platform-independence by using byte code. The Java compiler does not produce native executable code for a particular machine like a C compiler does. Instead it produces a special format called byte code. Java byte code written in hexadecimal, byte by byte, looks like this:
CA FE BA BE 00 03 00 2D 00 3E 08 00 3B 08 00 01 08 00 20

Platform Independent



Platform Independent

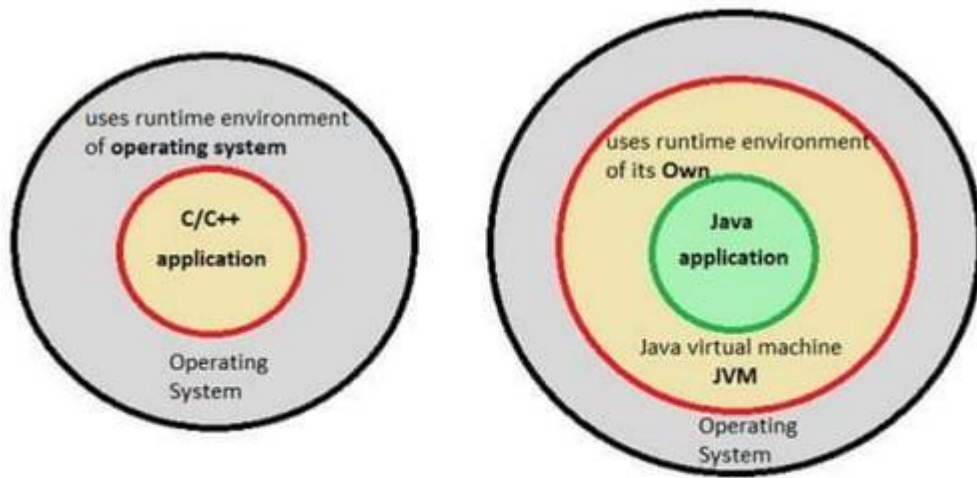


Features of Java

- **Secured**
 - Java is a secure programming language because:
 - No explicit pointer
 - Program run inside virtual machine sandbox.
 - Array index limit checking
 - Code pathologies reduced by :
 - **Bytecode verifier** – Checks classes after loading (code fragments for illegal code that can violate access right to object.)
 - **ClassLoader** – confines objects to unique namespaces. Prevents loading a hacked “java.lang.SecurityManager” class.
 - **Security manager** – determines what resource a class can access such as reading and writing to the local disk.

Secured

Program run inside virtual machine sandbox



Features of Java

- **Robust**

- Robust simply means strong. Java uses strong memory management. There are lack of pointers that avoids security problem.
- There is automatic garbage collection in java.
- There is exception handling and type checking mechanism in java.
- All these points makes java robust.

Features of Java

- **Architecture-Neutral**

- There is no implementation dependent features e.g. 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. But in java, it occupies 4 bytes of memory for both 32 and 64 bit architectures.
- A Language or Technology is said to be Architectural neutral which can run on any available processors in the real world without considering there architecture and vendor (providers) irrespective to its development and compilation.

Features of Java

- **High performance**
- It have high performance because of following reasons;
- This language **uses Bytecode** which is more faster than ordinary pointer code so Performance of this language is high.
- **Garbage collector**, collect the unused memory space and improve the performance of application.
- It have **no pointers** so that using this language we can develop an application very easily.
- It **support multithreading**, because of this time consuming process can be reduced to execute the program.

Features of Java

- **Networked**

- It is mainly design for web based applications, J2EE is used for developing network based applications.

- **Dynamic**

- It support Dynamic memory allocation due to this memory wastage is reduce and improve performance of application. The process of allocating the memory space to the input of the program at a run-time is known as dynamic memory allocation, To programming to allocate memory space by dynamically we use an operator called 'new' 'new' operator is known as dynamic memory allocation operator.

Features of Java

- **Distributed**

- We can create distributed applications in java. RMI and EJB are used for creating distributed applications. We may access files by calling the methods from any machine on the internet.

- **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.

Centralized Applications

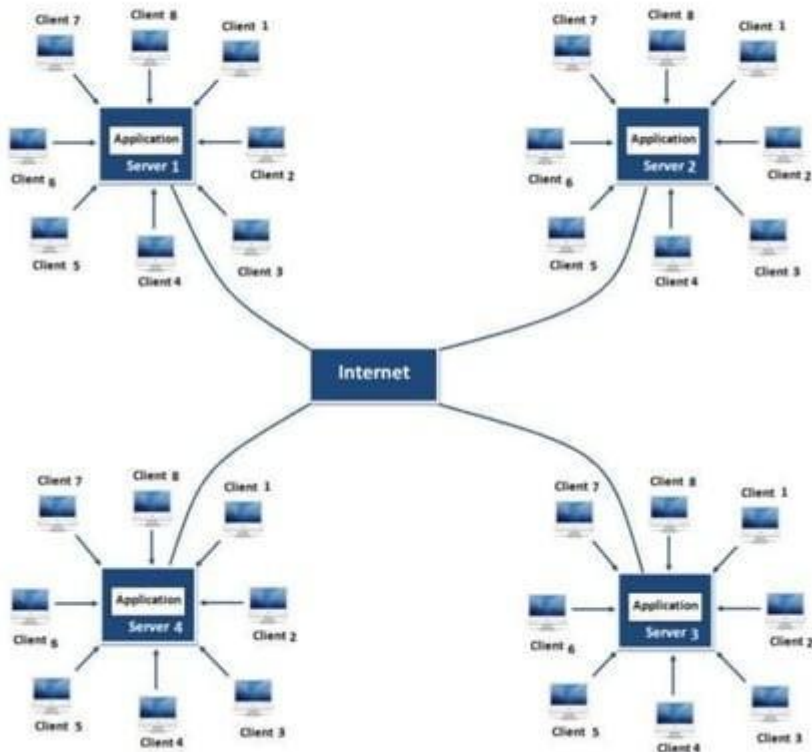
- In this scenario multiple clients system depends on single server system.
- The major drawback in this architecture is if any problem occurred on server system that will be reflected on every client system.



Distributed Applications

- In this scenario multiple client system are depends on multiple server systems so that even problem occurred in one server will never be reflected on any client system.
- Java is very powerful language can be used to developed both client server architecture and distributed architecture based application.
- **Note:** In this architecture same application is distributed in multiple server system.

Distributed Applications



Features of Java

- **What is Byte Code in java**
- The code which is converted by java compiler(**javac**) is called **byte code** in java.
- **Why Java is Portable?**
- The code which is converted by java compiler(**javac**) is non executable.
- **Byte code** is a highly optimized set of instructions
- **Byte code** is executed by Java run-time system, which is called the **java virtual machine (JVM)**.
- **Byte code** is **intermediate language** of source code and **executable** code.
- JVM converts byte code of java into machine language to execute **microprocessor** of OS.
- However internal details of **JVM** will differ from platform to platform but still all **JVM** understands the same java **bytecode**. Or in other word we can say **JVM** is platform dependent and **java byte code** is **platform independent**.

Basic Syntax of Java Program

- About Java programs, it is very important to keep in mind the following points.
- **Case Sensitivity** - Java is case sensitive, which means identifier **Hello** and **hello** would have different meaning in Java.
- **Class Names** - For all class names the first letter should be in Upper Case.

If several words are used to form a name of the class, each inner word's first letter should be in Upper Case.

Example ***class** MyFirstJavaClass*

Basic Syntax of Java Program

- **Method Names** - All method names should start with a Lower Case letter.
- If several words are used to form the name of the method, then each inner word's first letter should be in Upper Case.
- Example ***public void** myMethodName()*

Basic Syntax of Java Program

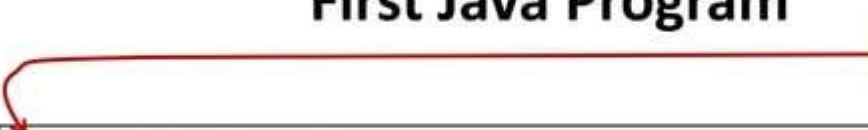
- **Program File Name** - Name of the program file should exactly match the class name.

When saving the file, you should save it using the class name (Remember Java is case sensitive) and append '.java' to the end of the name (if the file name and the class name do not match your program will not compile).

Example: Assume 'MyFirstProgram' is the class name. Then the file should be saved as *'MyFirstProgram.java'*

First Java Program


Class Keyword



```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```


First Java Program

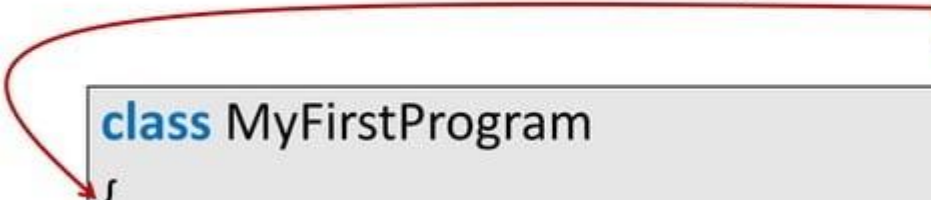
Class Name



```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```


First Java Program

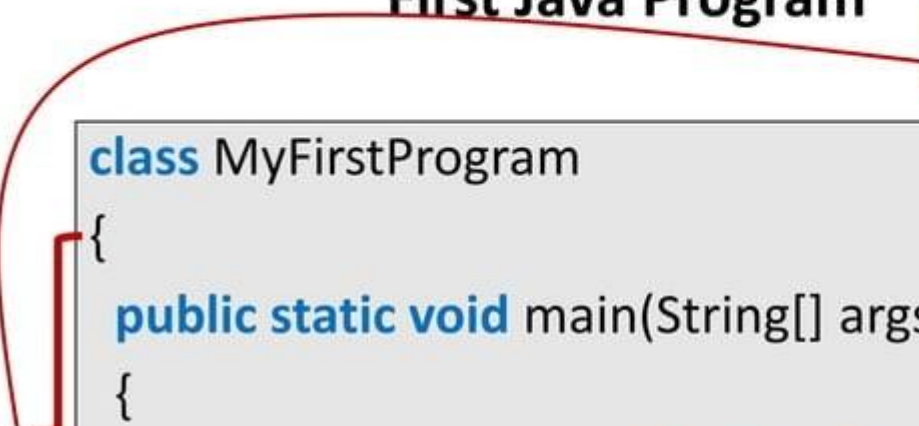
Starting of Class
Level Scope



```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```

First Java Program

This is Called
One Block



```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```

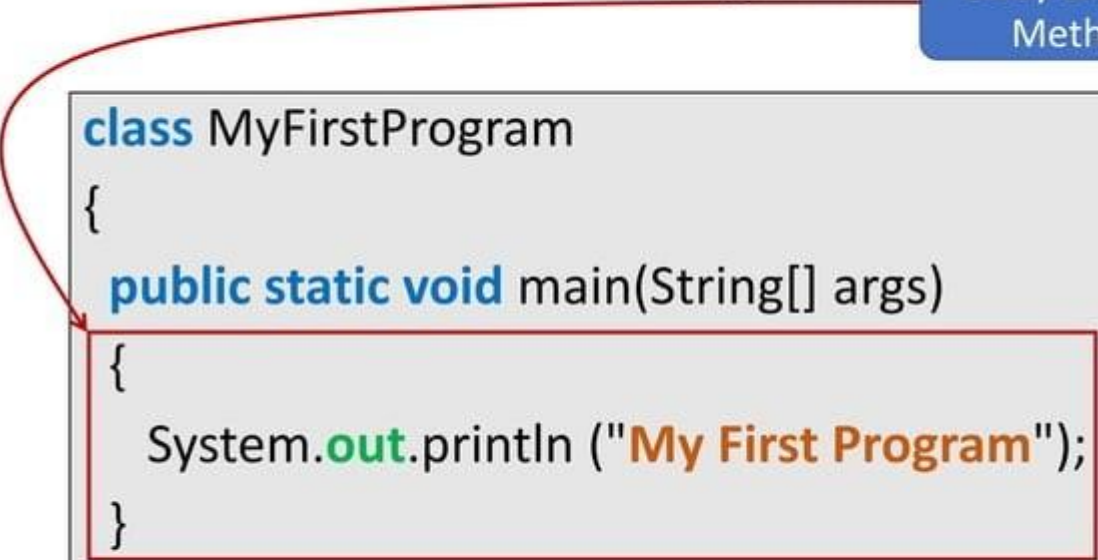
First Java Program

This is Called
Main Method

```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```

First Java Program


Body of Main
Method



```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```

First Java Program

Start of Main
Method level
scope



```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```

First Java Program

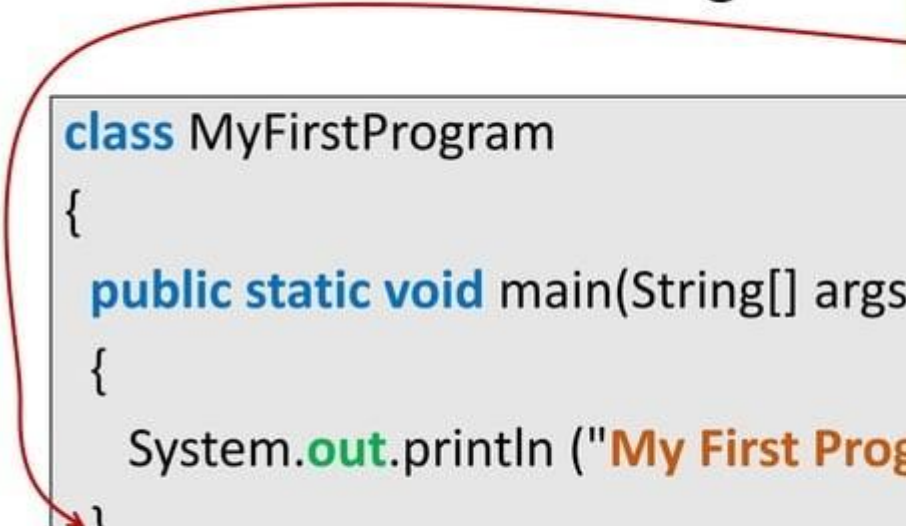
Print on Screen

```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```



First Java Program

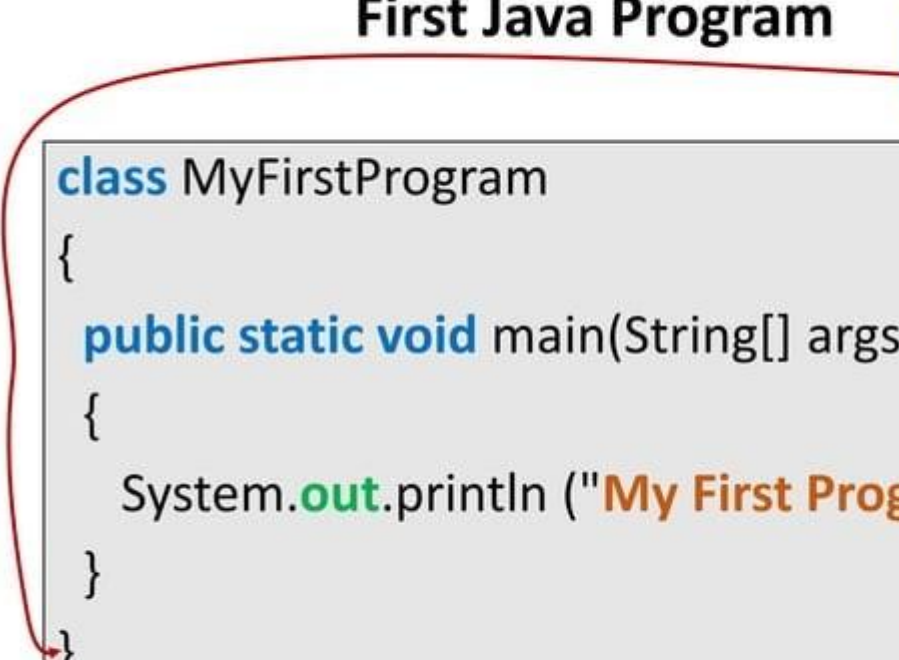
End of Main
Method



```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```

First Java Program


End of Class



```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```


First Java Program

Semicolon use
to terminate
the statement.




```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```

Blocks in Java

- A *block* (or a **compound statement**) is a group of statements surrounded by **braces** { }.
- **All the statements** inside the block is **treated as one unit**. Blocks are used as the *body* in constructs like function, if-else and loop, which may contain multiple statements but are treated as one unit.
- **For example**

```
public static void main(String[] args)
{
    System.out.println ("My First Program");
}
```



This is one
block

- **class** – is use to declare a class in java.
- **public** – is a access modifies which represent visibility. public means it is visible to everyone.
- **static** – is a keyword. If we any method as static , it is known as static method. The advantage of static method is that there is no need to create object to invoke the static method. The main method is executed by the JVM, so it does not require to create object to invoke the main method. So it saves memory.
- **void** – is the return type of the method. void means method does not return any value.
- **main** – represent startup of the program.
- **String args[]** – is used for command line argument. we will learn it later.
- **System.out.println()** – is used to print statement.

Internal Details of Java First Program

- What happens at compile time?
- At compile time, java file is compiled by Java Compiler (It does not interact with OS) and converts the java code into bytecode.

```
class MyFirstProgram
{
    public static void main(String[] args)
    {
        System.out.println ("My First Program");
    }
}
```

MyFirstProgram.java



Compiler



Bytecode

MyFirstProgram.class

Tools We Will Need

- **For this Course We will need the following software's:**
- Linux 7.1 or Windows XP/7/8/10 Operating System.
- Java JDK 8
- NetBeans 8.1

Download Java Development Kit

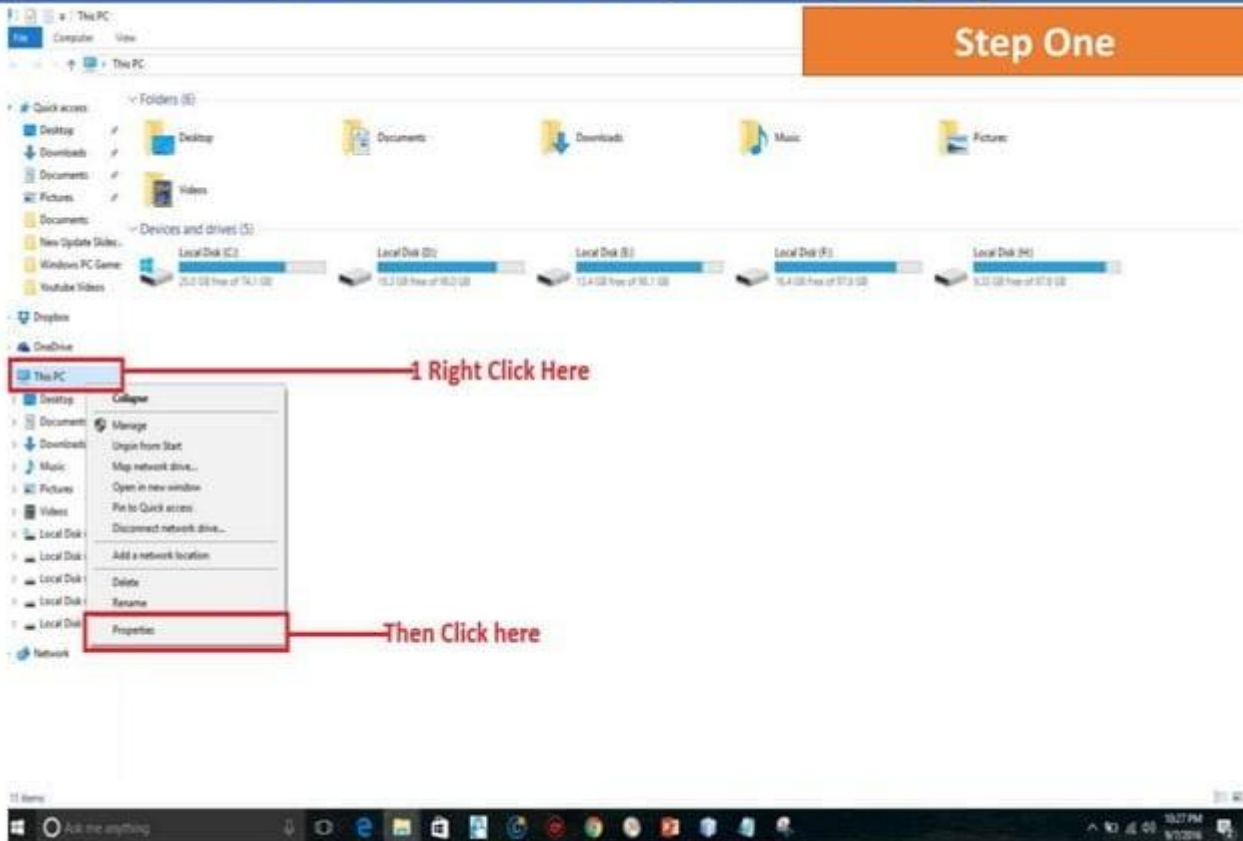
- **Download Java SE Development Kit 8 According to your Operating System *architecture (32bit or 64bit)***
- <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- We can also download JDK from any site with latest version probably version 8 is latest.

Please Note

- If you don't know about your **Operating System Architecture** is it 32bit or 64bit so follow me to find this..
- Go on my computer icon and right click, after that click on properties option

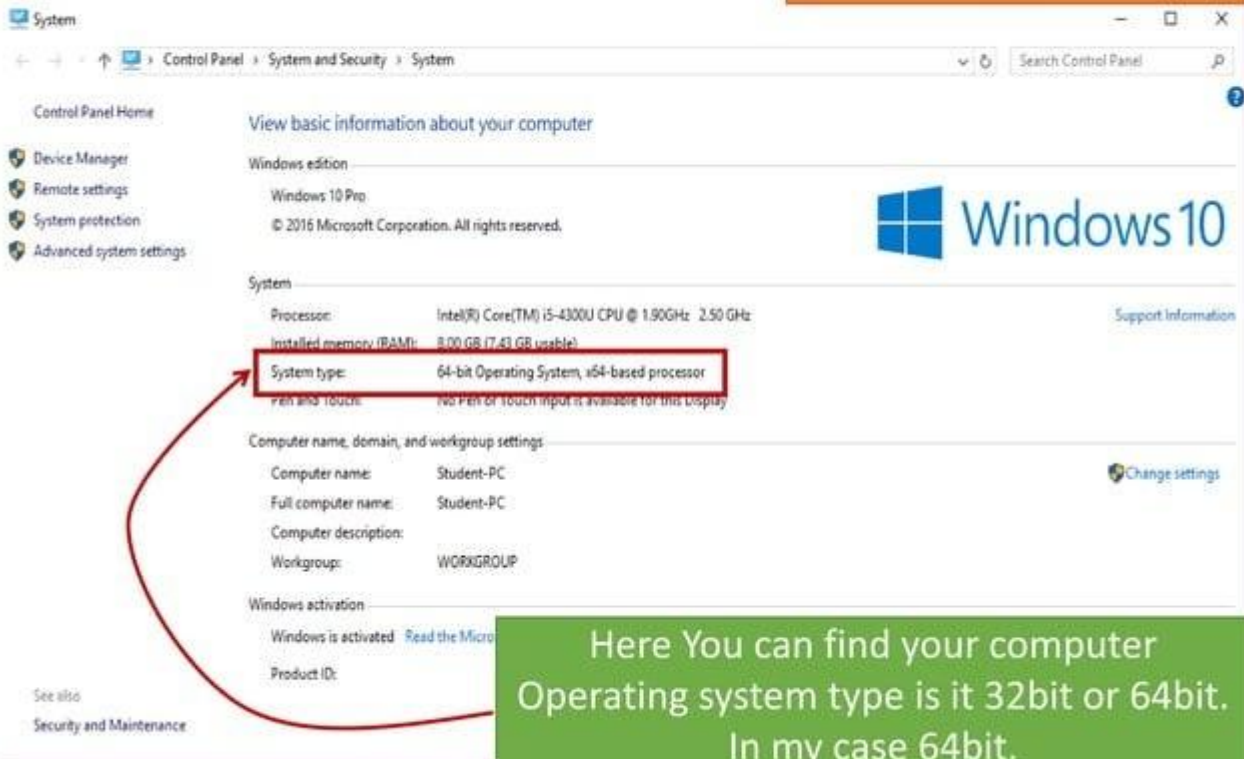
Introduction to Java Programming Language

Step One



Introduction to Java Programming Language

Step Two



The screenshot shows the Windows 10 'System' settings window. The left sidebar contains links to 'Control Panel Home', 'Device Manager', 'Remote settings', 'System protection', and 'Advanced system settings'. The main content area is titled 'View basic information about your computer' and displays the following information:

- Windows edition:** Windows 10 Pro, © 2016 Microsoft Corporation. All rights reserved.
- System:**
 - Processor: Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz 2.50 GHz
 - Installed memory (RAM): 8.00 GB (7.43 GB usable)
 - System type: 64-bit Operating System, x64-based processor** (highlighted with a red box)
 - Pen and touch: No Pen or touch input is available for this display
- Computer name, domain, and workgroup settings:**
 - Computer name: Student-PC
 - Full computer name: Student-PC
 - Computer description:
 - Workgroup: WORKGROUP
- Windows activation:**
 - Windows is activated [Read the Microsoft software license terms](#)
 - Product ID:

A red arrow points from the 'System type' entry to a green callout box at the bottom right. The callout box contains the text: 'Here You can find your computer Operating system type is it 32bit or 64bit. In my case 64bit.'