

## UNIT-1 (Introduction to Java)

Installing Visual Studio Code(thomasbhattarai@gmail.com)

Installing Java in Visual Studio Code.

1. Go to <https://code.visualstudio.com/docs/languages/java>

2. *Download Package*

## Install Visual Studio Code for Java

To help you set up quickly, we recommend you use the **Coding Pack for Java**, which is the bundle of VS Code, the Java Development Kit (JDK), and a collection of suggested extensions by Microsoft. The Coding Pack can also be used to fix an existing development environment.

Install the Coding Pack for Java - Windows

Install the Coding Pack for Java - macOS

**Note:** The Coding Pack for Java is only available for Windows and macOS. For other operating systems, you will need to manually install a JDK, VS Code, and Java extensions.

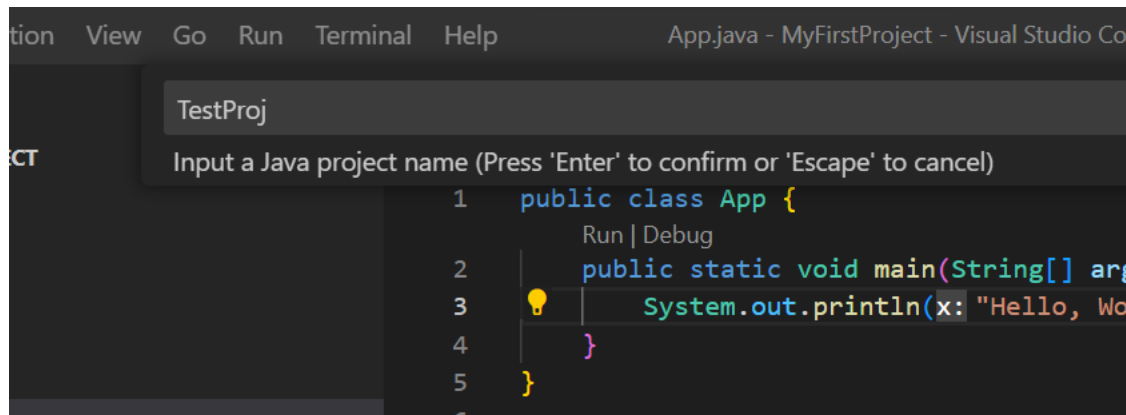
3. Open VS Code. Now open command ctrl+shift+p (in VS Code)

4. Select in dropdown (Java:create java project)

5. Select (No built tools)

6. Select your Folder Location

7. Type name of your project given in screenshot



## Introduction:

- Java is a **high-level, object-oriented** programming language developed by **Sun Microsystems** (now owned by **Oracle Corporation**). It was first released in 1995 and has since become one of the most popular and widely used programming languages for building a variety of **software applications**, including **web, mobile, enterprise, and desktop applications**.
- Key features of Java include **platform independence**, which means that Java programs can run on any device with a Java Virtual Machine (JVM).
- Java follows the "**Write Once, Run Anywhere**" (WORA) principle, allowing developers to write code once and run it on different platforms without modification.
- Java also has strong **community support**, an extensive ecosystem of **libraries** and **frameworks**, and it is widely **used in enterprise** environments for building robust, scalable, and secure applications.

## Features of Java:

### Platform Independence:

Java achieves platform independence by compiling code into **bytecode**, allowing it to run on any device with a **Java Virtual Machine (JVM)**, following to the "**Write Once, Run Anywhere**" (WORA) principle.

### Object Oriented:

Java is object-oriented, promoting the use of classes and objects. It supports principles like **encapsulation, inheritance, abstraction** and **polymorphism**, making it easy to **design** and maintain **modular** and **extensible** code.

### Simple and Easy to Learn:

Java was designed to be user-friendly and accessible to developers. Its syntax is similar to other **C-based** languages, and it avoids complex features, making it relatively easy for beginners to learn.

### **Robust and Secure:**

Java incorporates features to enhance program robustness, such as **automatic memory management** (garbage collection) and **strong type-checking**. The Java Virtual Machine provides a **secure execution environment**, minimizing the **risk of memory-related errors** and **security vulnerabilities**.

### **Multi-threading:**

Java supports multithreading, allowing concurrent execution of multiple threads within a program.

### **Rich Standard Library:**

Java comes with a comprehensive standard library that provides **pre-built classes** and **packages** for common tasks. This library includes utilities for **input/output, data structures, networking, and more**, saving developers time and effort.

### **Distributed:**

Java designed for distributed internet environments as it manages the TCP/IP Protocol

### **History of JAVA:**

Java is the general purpose, **true object oriented programming language** and is highly suitable for modeling the real world and solving the real world problems. In the company **Sun Microsystem**, **James Goslings**(an employee there), started a project named **“GREEN”**. He begun his work by attempting to extend C++ features for developing embedded software for **electronic companies** to automate the **electronic devices** such as microwave. But this could not be accomplished by C++.

So **James Goslings** started to develop a new language known as **“OAK”**. In may 1995, Sun officially announced this language at **Sunworld 95**. Due to some reasons, the name was changed to **“JAVA”**.

The prime motive of Java was the need for a **platform independent** language. The second motive was to design a language which is **“Internet Enable”**. To run the programs on internet, Green project team come up with the idea of developing **“Web Applet”**. The team developed a **web browser called “Hot Java”** to locate and run applet programs on internet.

The most striking feature of Java is **“Platform neutrality”**. Java is the first language that is not tied to any particular hardware or O.S.

## **JAVA MILESTONES**

YEAR	DEVELOPMENT
1990	Sun Microsystem started to develop special software for electronic devices.
1991	James Goslings developed a new language “OAK”.
1992	New language Oak was demonstrated in hand held devices.
1993	World Wide Web(WWW)appeared on internet and transformed the text based internet to graphical environment. Sun developed Applets.
1994	Green project team developed a web browser “Hot Java” to run applets on internet.
1995	Oak was renamed to JAVA.
1996	Java established itself as the leader of internet programming. Sun released Java development kit 1.0.
1997	Sun released Java development kit 1.1(JDK 1.1).
1998	Sun released Java2 with version 1.2(SDK 1.2).
1999	Sun released Java2 standard edition (J2SE) and Enterprise Edition (J2EE).
2000	J2SE with SDK 1.3 was released.
2002	J2SE with SDK 1.4 was released.
2004	J2SE with JDK 5.0 was released.

### **The Internet and Java’s Place in IT:**

Earlier Java was only used to design and program small computing devices, but it was later adopted as one of the platform-independent programming languages, and now according to Sun, 3 billion devices run Java. Java is one of the most important programming languages in today's IT industries.

**JSP** - In Java, JSP (Java Server Pages) is used to create dynamic web pages, such as in PHP and ASP.

**Applets** - Applets are another type of Java program that are implemented on Internet browsers and are always run as part of a web document.

**J2EE** - Java 2 Enterprise Edition is a platform-independent environment that is a set of different protocols and APIs and is used by various organizations to transfer data between each other.

**JavaBeans** - This is a set of reusable software components that can be easily used to create new and advanced applications.

**Mobile** - In addition to the above technology, Java is widely used in mobile devices nowadays, and many types of games and applications are being made in Java

## Types of Java Application and Applets:

There are mainly 4 type of applications that can be create using java.

Basically we divide java based on in it character and we can consider hardware in categorization of java that is below:-

### 1) Standalone Application

It is also known as **desktop application or window-based application**. An application that we need to install on every machine such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.

### 2) Web Application

An application that runs on the server side and creates dynamic page, is called web application. Currently, **servlet, jsp** etc. technologies are used for creating web applications in java.

### 3) Enterprise Application

An application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

### 4) Mobile Application

An application that is created for mobile devices. Currently Android and Java ME are used for creating mobile applications.

## Java Virtual Machine(JVM):

The Java Virtual Machine (JVM) serves as a **runtime environment** for executing Java **bytecode**, adhering to a standardized specification. As the name suggests, the JVM operates as a **virtual processor**, enabling **platform-independent** execution of Java programs.

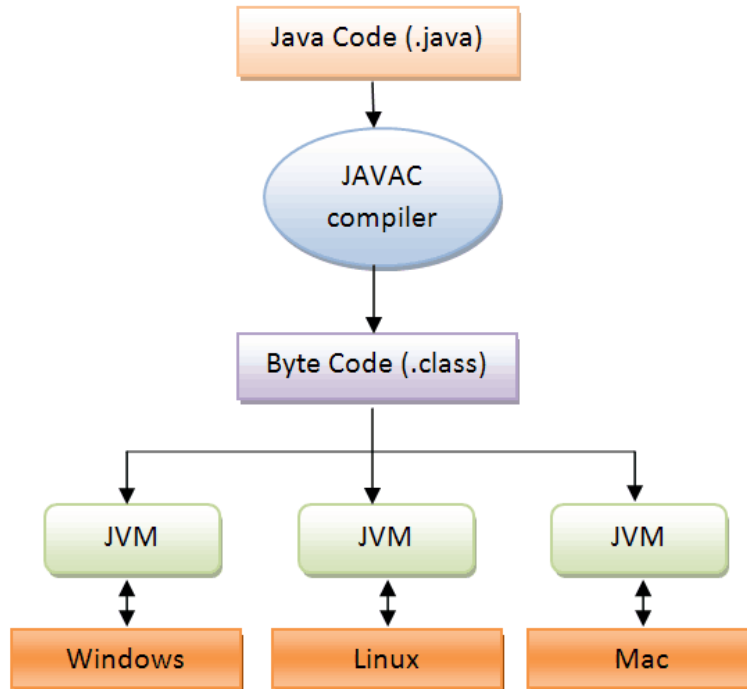
The JVM performs following operation:

- Loads code
- Verifies code
- Interpreting
- Executes code

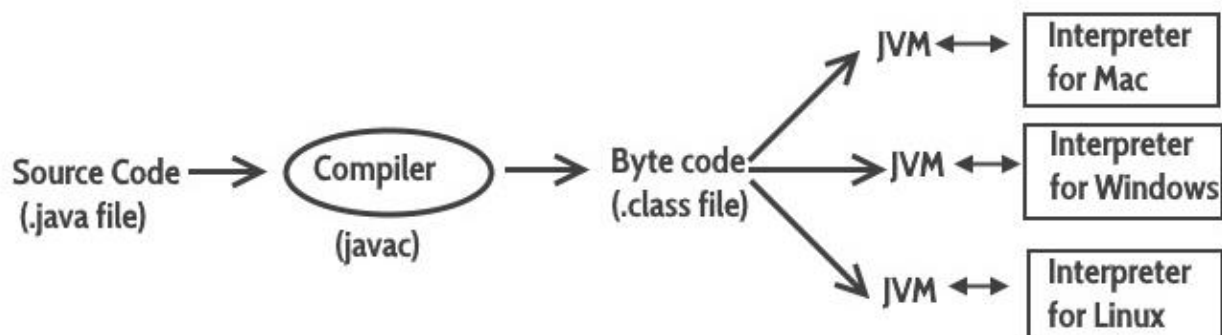
The JVM takes on the crucial tasks of loading class files, verifying the code, interpreting it, and executing it.

Java is a high level programming language. A program written in high level language cannot be run on any machine directly. First, it needs to be translated into that particular machine language. The **javac compiler** does this thing, it takes java program (.java file containing source code) and translates it into machine code (referred as byte code or .class file).

Java Virtual Machine (JVM) is a virtual machine that resides in the real machine (your computer) and the **machine language for JVM is byte code**. This makes it easier for compiler as it has to generate byte code for JVM rather than different machine code for each type of machine. JVM executes the byte code generated by compiler and produce output. JVM is the one that makes **java platform independent**.



This design approach enables Java programs to be **platform-independent**, as the same **bytecode** can be executed by any JVM on various operating systems, ensuring the "Write Once, Run Anywhere" principle of Java.



**Fig: Compilation and Interpretation process**

The JVM is responsible for executing Java bytecode. It translates bytecode into machine-specific instructions, allowing Java programs to run on various platforms without modification.

## Procedural Programming Language vs Object Oriented Programming

Procedural Oriented Programming	Object-Oriented Programming
In procedural programming, the program is divided into small parts called <b>functions</b> .	In object-oriented programming, the program is divided into small parts called <b>objects</b> .
Procedural programming follows a <b>top-down approach</b> .	Object-oriented programming follows a <b>bottom-up approach</b> .
There is <b>no access specifier</b> in procedural programming.	Object-oriented programming has access specifiers like <b>private, public, protected</b> , etc.
Adding new data and functions is not easy.	Adding new data and function is easy.
Procedural programming does not have any proper way of hiding data so it is <b>less secure</b> .	Object-oriented programming provides data hiding so it is <b>more secure</b> .
In procedural programming, <b>overloading is not possible</b> .	<b>Overloading is possible</b> in object-oriented programming.
In procedural programming, there is <b>no concept of data hiding and inheritance</b> .	In object-oriented programming, <b>the concept of data hiding and inheritance</b> is used.
In procedural programming, <b>the function is more important than the data</b> .	In object-oriented programming, <b>data is more important than function</b> .
Procedural programming is based on the unreal world.	Object-oriented programming is based on the real world.
Procedural programming is used for <b>designing medium-sized programs</b> .	Object-oriented programming is used for <b>designing large and complex programs</b> .
Procedural programming uses the concept of <b>procedure abstraction</b> .	Object-oriented programming uses the concept of <b>data abstraction</b> .
<b>Code reusability absent</b> in procedural programming,	<b>Code reusability present</b> in object-oriented programming.
Examples: C, FORTRAN, Pascal, Basic, etc.	Examples: C++, Java, Python, C#, etc.

### Compiling and Running Simple Java Program:

- Open **notepad** write code shown below

```
public class App {  
    public static void main(String[] args) {  
        System.out.print("Hello");  
        System.out.println("Good morning");  
    }  
}
```

- Save this program with the name App.java.
- Click on start menu and go to run option and type cmd.
- Now command prompt will be opened.
- Go to the location where program is saved.
- Set the path of that location in command prompt.
- Give command to compile the program: **javac App.java**
- Give the command to run the program. : **java App**