# Unit 1
# Introduction to Java

# Introduction to Java

- Java is a class-based, object-oriented programming language.
- A general-purpose programming language made for developers to *write once run anywhere* i.e. compiled Java code can run on all platforms that support Java
- JAVA was developed by Sun Microsystems Inc in 1991, later acquired by Oracle Corporation.
- Java applications are compiled to byte code that can run on any Java Virtual Machine.
- It is used for:
  - Mobile applications (specially Android apps)
  - Desktop applications
  - Web applications
  - Web servers and application servers
  - Games
  - Database connection

## Introduction to Java

- Java is a high level programming language characterized by following traits
  - Simple and Familiar
  - Object-Oriented
  - Robust and Secure
  - Architecture Neutral and Portable
  - High Performance
  - Multi-threaded, Interpreted and Dynamic

## Why Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful
- It has a huge community support (tens of millions of developers)
- Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs

# History of Java

- James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. The small team of sun engineers called Green Team.
- Initially it was designed for small, embedded systems in electronic appliances like set-top boxes.
- Firstly, it was called **"Greentalk"** by James Gosling, and the file extension was **.gt**. After that, it was called **Oak** and was developed as a part of the Green project. Oak is a symbol of strength and chosen as a national tree of many countries like the U.S.A., France, Germany, Romania, etc.
- In 1995, Oak was renamed as **"Java"** because it was already a trademark by Oak Technologies. Java is an island in Indonesia where the first coffee was produced (called Java coffee). It is a kind of espresso bean. Java name was chosen by James Gosling while having a cup of coffee nearby his office.
- In 1995, Time magazine called **Java one of the Ten Best Products of 1995**.

# History of Java

- JDK 1.0 was released on January 23, 1996. After the first release of Java, there have been many additional features added to the language.
- Now Java is being used in Windows applications, Web applications, enterprise applications, mobile applications, cards, etc. Each new version adds new features in Java.

## Java Applications

1. Standalone Application
   - desktop / window base applications that needs to be installed in every machine
   - AWT(Abstract Window Toolkit) and Swing are used in Java for creating standalone applications
2. Web Application
   - an application that runs on server side and creates dynamic pages
   - JSP(Java Server Pages), Spring, Hibernate etc. are used to create web applications in Java
3. Enterprise Application
   - have distributed nature, such as banking application
   - EJB(Enterprise Java Bean) is used for creating enterprise application in Java
4. Mobile Application
   - created for mobile devices (android)

# Applets in Java

- An applet is a Java program that can be embedded into a web page. It runs inside the web browser and works at client side.
- An applet is embedded in an HTML page using the APPLET or OBJECT tag and hosted on a web server.
- Applets are used to make the website more dynamic and entertaining.
- Advantages:
  - It works at client side so less response time.
  - Secured
  - It can be executed by browsers running under many platforms, including Linux, Windows, Mac Os etc.

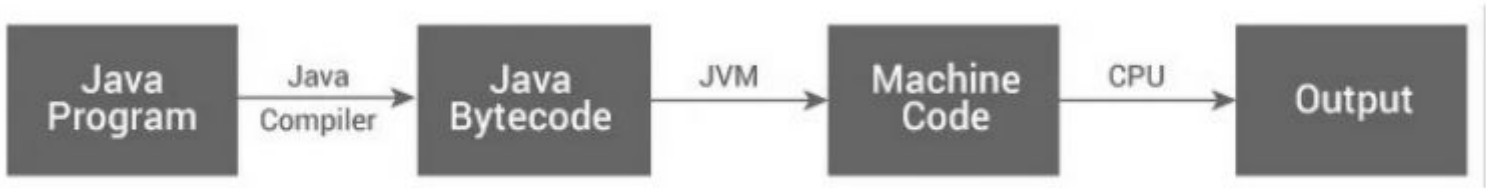
# The Internet and Java

- Java is strongly associated with the internet
- Java influenced Internet by simplifying the web programming and inventing applets.
  - An applet is a special kind of program that is transmitted over the Internet and automatically executed by the Java-compatible web browser.
  - Applets are usually small programs and these helped in moving some user interactive programs from server to client, hence improving the usability of the web application.
- Security: Since applets are automatically downloaded and run on the client machine, there are restrictions on what can be done in Applets. They can use only a subset of all the functions supported by Java.
- Portability: Applets makes program portable

# Java JDK, JRE and JVM

## Java Virtual Machine(JVM)

- JVM is an abstract machine that enables the computer to run a Java program
- Java compiler first compiles our Java code (file with .java extension) to bytecode(file with .class extension).
- Then, the JVM translates bytecode into native machine code (set of instructions that a computer's CPU executes directly)
- Java is a platform-independent language. It's because when we write Java code, it's ultimately written for JVM but not the physical machine (computer).
- Since JVM executes the Java bytecode which is platform-independent, Java is platform-independent



# Java JDK, JRE and JVM

## Java Runtime Environment(JRE)

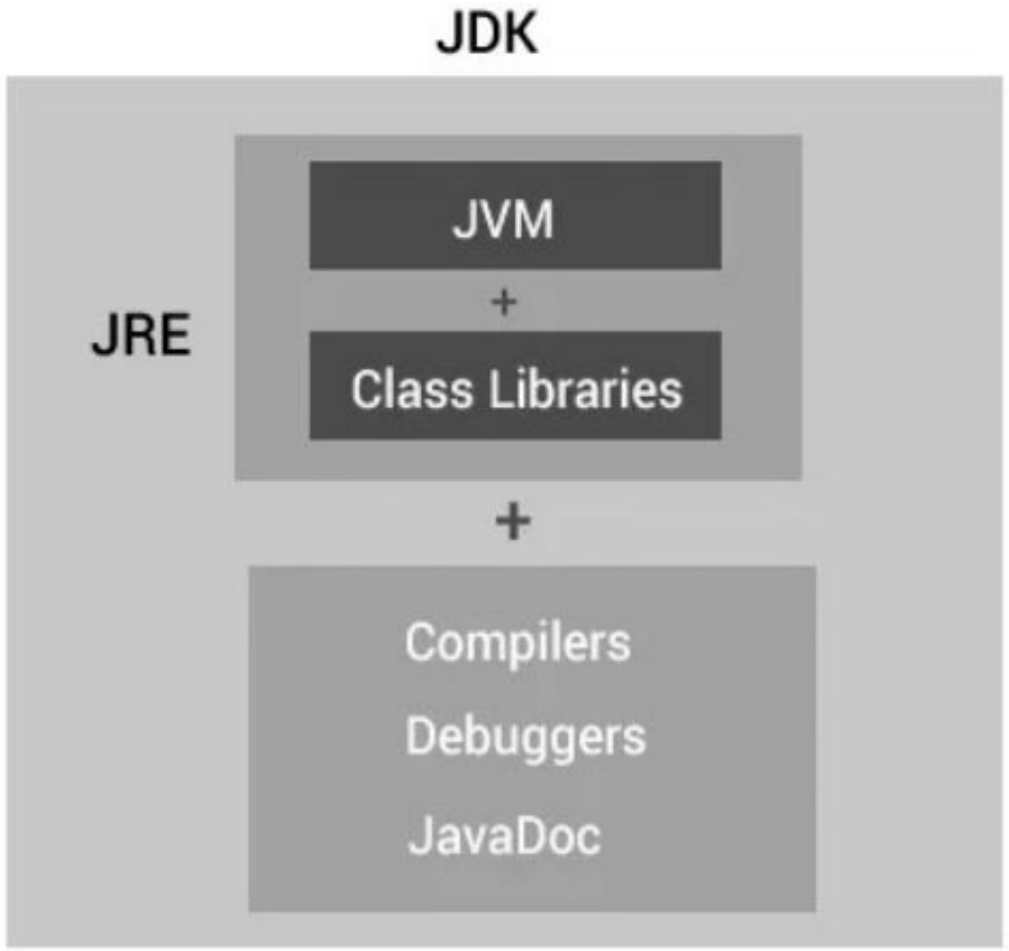
- JRE is a software package that provides Java class libraries, Java Virtual Machine (JVM), and other components that are required to run Java applications
- If we need to run Java programs, but not develop them, JRE is what we need

# Java JDK, JRE and JVM

## Java Development Kit (JDK)

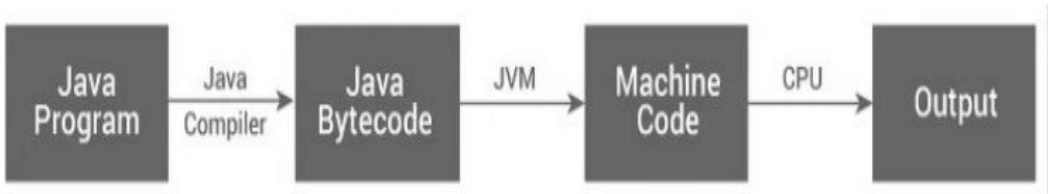
- JDK is a software development kit required to develop applications in Java.
- When we download JDK, JRE is also downloaded with it.
- In addition to JRE, JDK also contains a number of development tools (compilers, Javadoc, Java Debugger, etc.).

# Java JDK, JRE and JVM



# Byte Code in Java

- Byte Code can be defined as an intermediate code generated by the compiler after the compilation of source code(JAVA Program).
- This intermediate code makes Java a platform-independent language.



## Object Oriented Programming

- Object-oriented programming (OOP) is a programming paradigm based on the concept of objects and classes to organize the data

## Objects

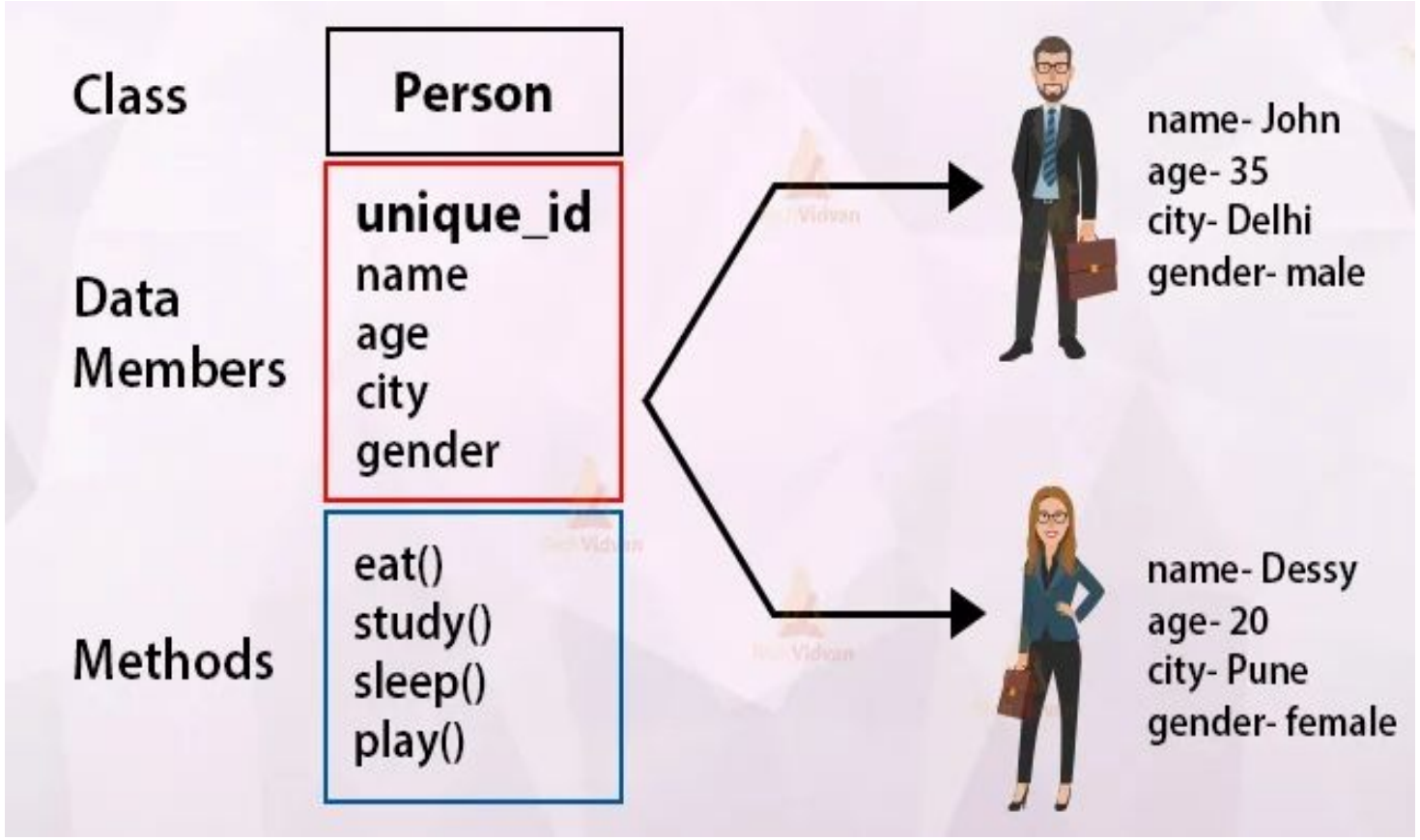
- Objects are the real world entities
- An object consists of :
  - **State**: It is represented by attributes of an object. It also reflects the properties of an object.
  - **Behavior**: It is represented by methods of an object. It also reflects the response of an object with other objects.
  - **Identity**: It gives a unique name to an object and enables one object to interact with other objects.

# Object Oriented Programming

## Class

- A class can also be defined as a blueprint from which you can create an individual object. Class doesn't consume any space.
- Collection of objects is called class. It is a logical entity.
- A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type.

# Object Oriented Programming-Class and Object



# Object Oriented Programming

❖ Besides concepts of classes and objects, OOP have following concepts

1. Inheritance
   - Inheritance is a mechanism in which one object acquires all the properties and behaviors of a parent object
2. Polymorphism
   - Polymorphism is a concept by which we can perform a single action in different ways
3. Abstraction
   - Abstraction is a process of hiding the implementation details and exposing only functionality
4. Encapsulation
   - Encapsulation is a process of wrapping code and data together into a single unit

# Procedure Oriented vs Object Oriented Programming

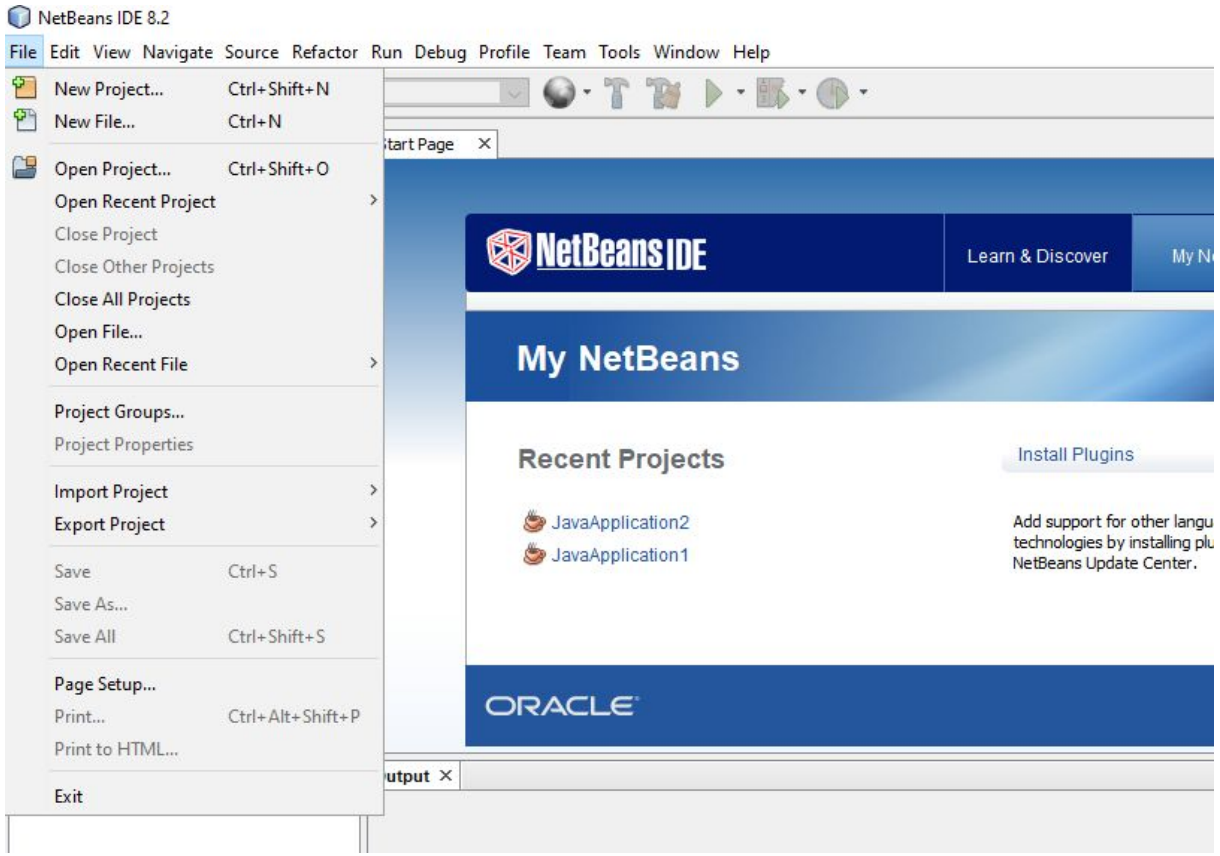
| Object Oriented Programming (OOP) | Procedure Oriented Programming(POP) |
|---|---|
| Program is divided into objects and their interactions. | Program is divided into functions. |
| **Bottom-up Approach** i.e. starting from solving small modules and adding them up together | **Top-down Approach** i.e. dividing a program into smaller parts and then solving it |
| Inheritance property is provided | Inheritance property is not allowed |
| It uses access specifiers | It doesn't use access specifiers. |
| Provides data hiding | No data hiding |
| Provides the feature of code reusability | No code reusability |
| Adding new data and functions is easy | Expanding new data and functions is comparatively difficult |
| Example: C++, Java | Example: C, Pascal |

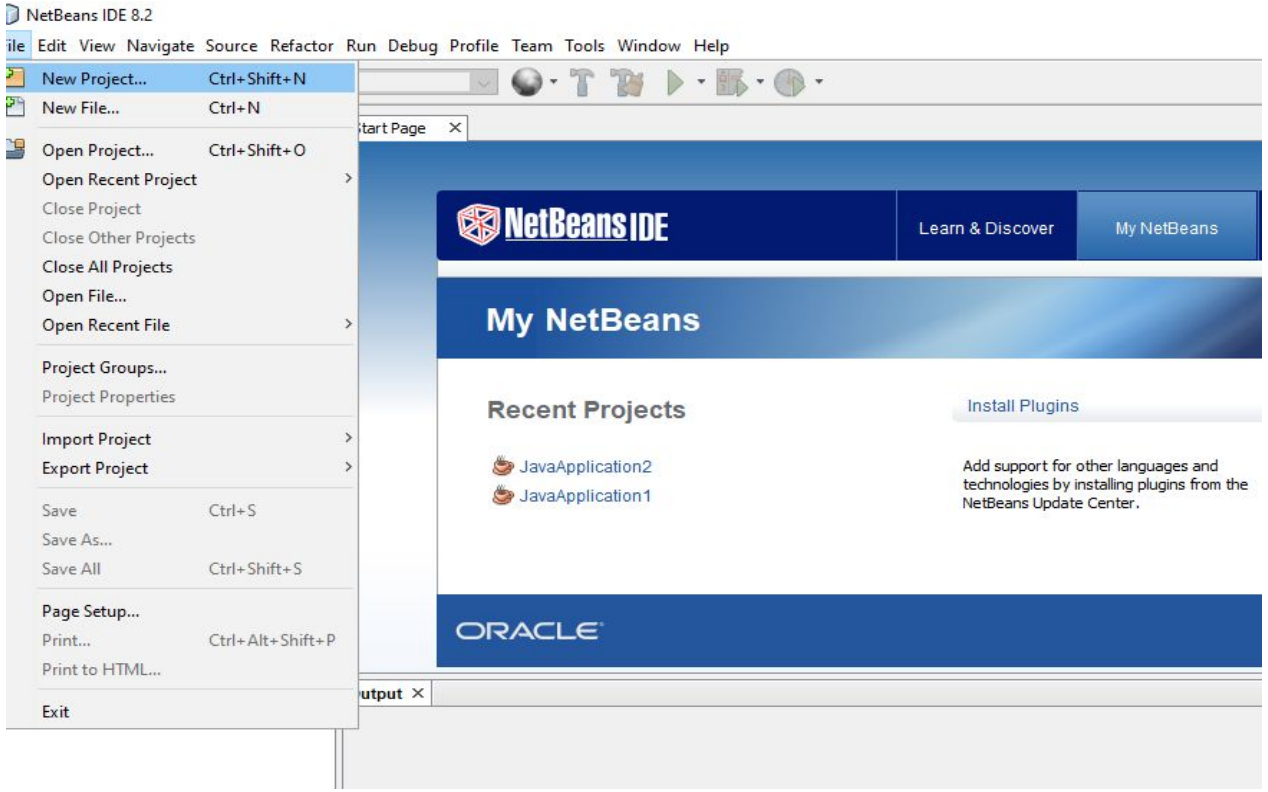
## Tools to start programming in Java

1. Install Java Development Kit (JDK) from oracle
2. Install an Integrated Development Environment(IDE)
   - Options:
     - Eclipse
     - IntelliJ IDEA
     - NetBeans
3. Test your installed software.

# Creating New Project in NetBeans

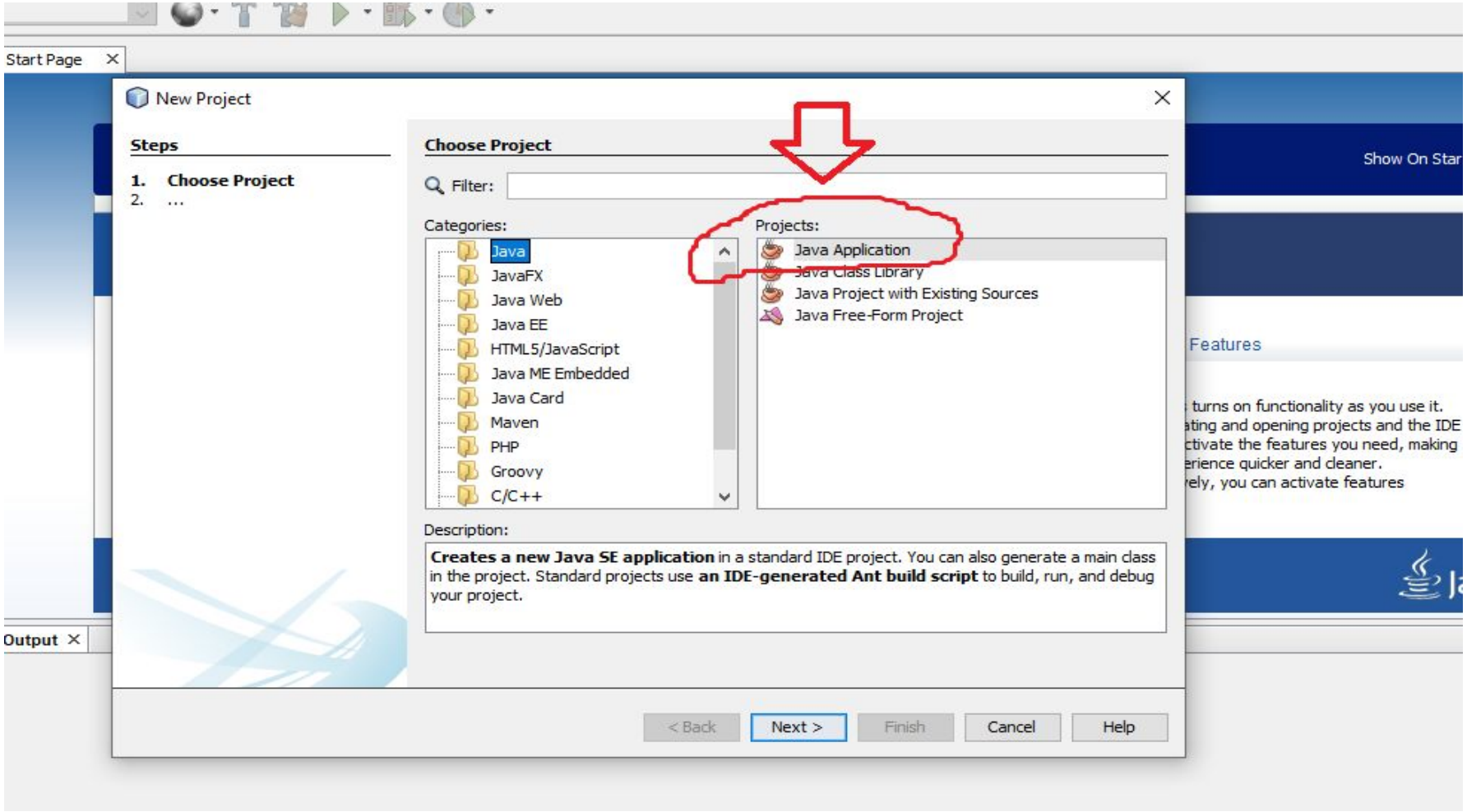
❖ Select New Project



# Creating New Project in NetBeans

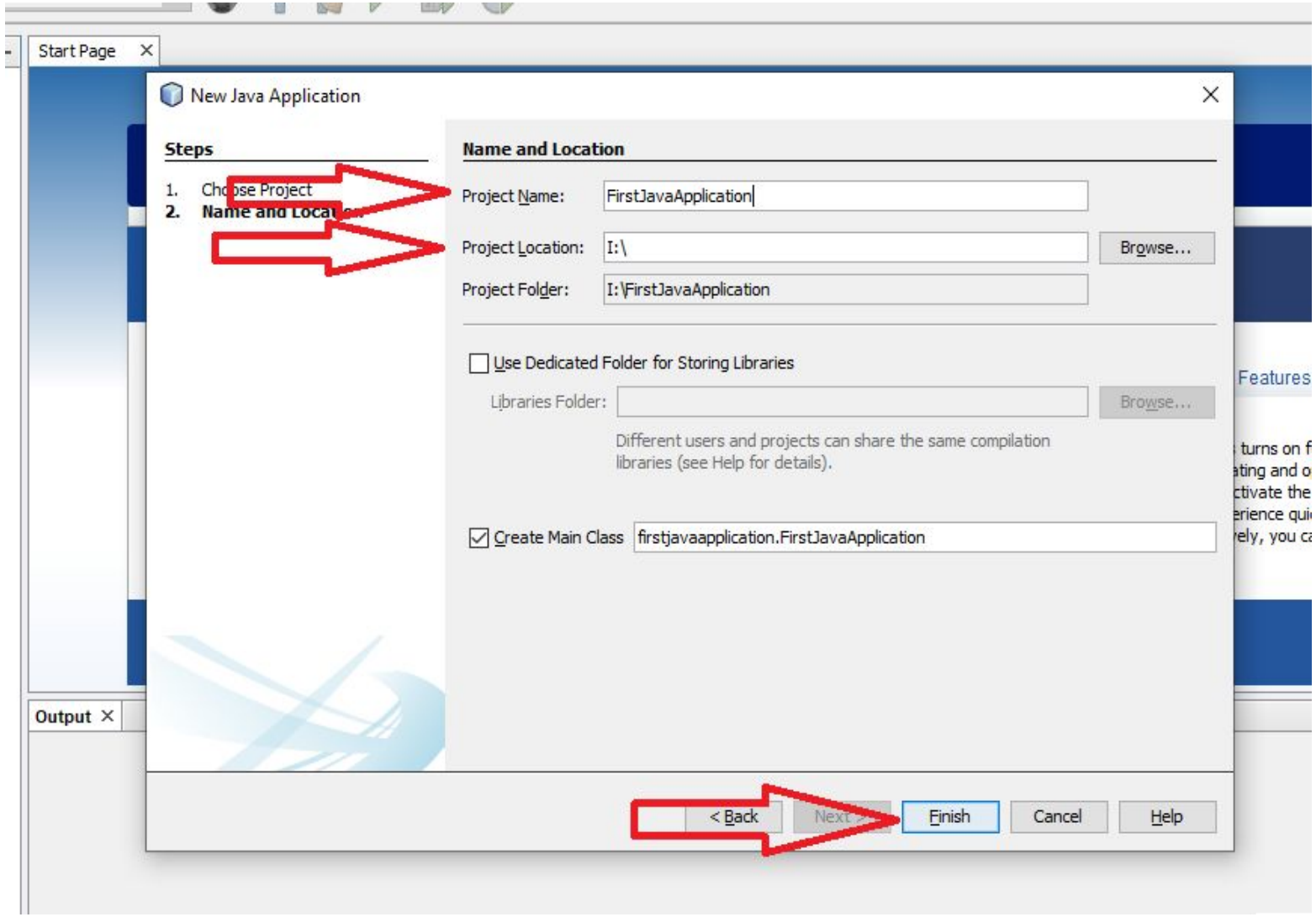
❖ Now a pop appears and from that select as below:




❖ Now select next

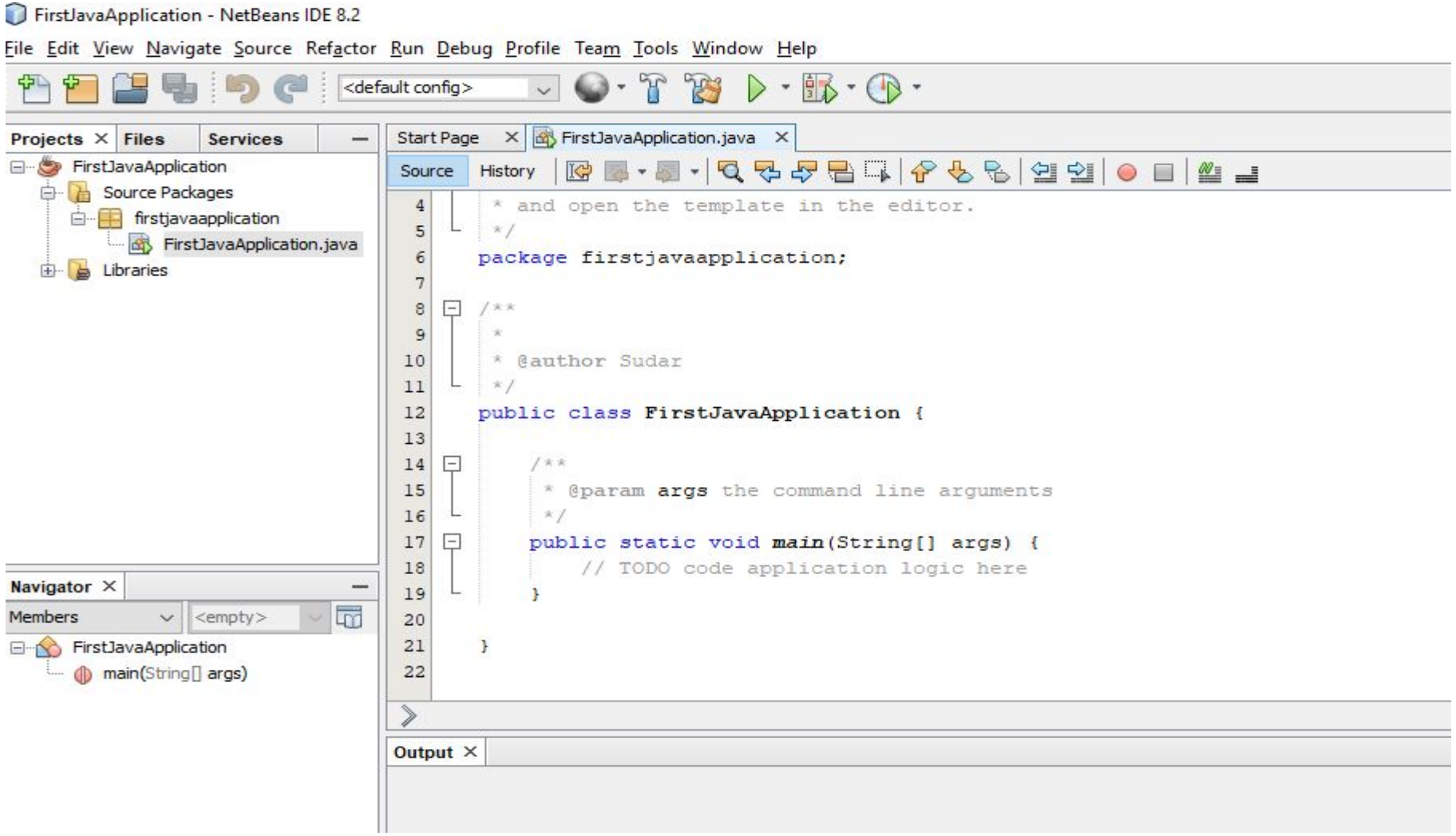
# Creating New Project in NetBeans

- Now enter the project name and select the location where you want to save the project and click finish.



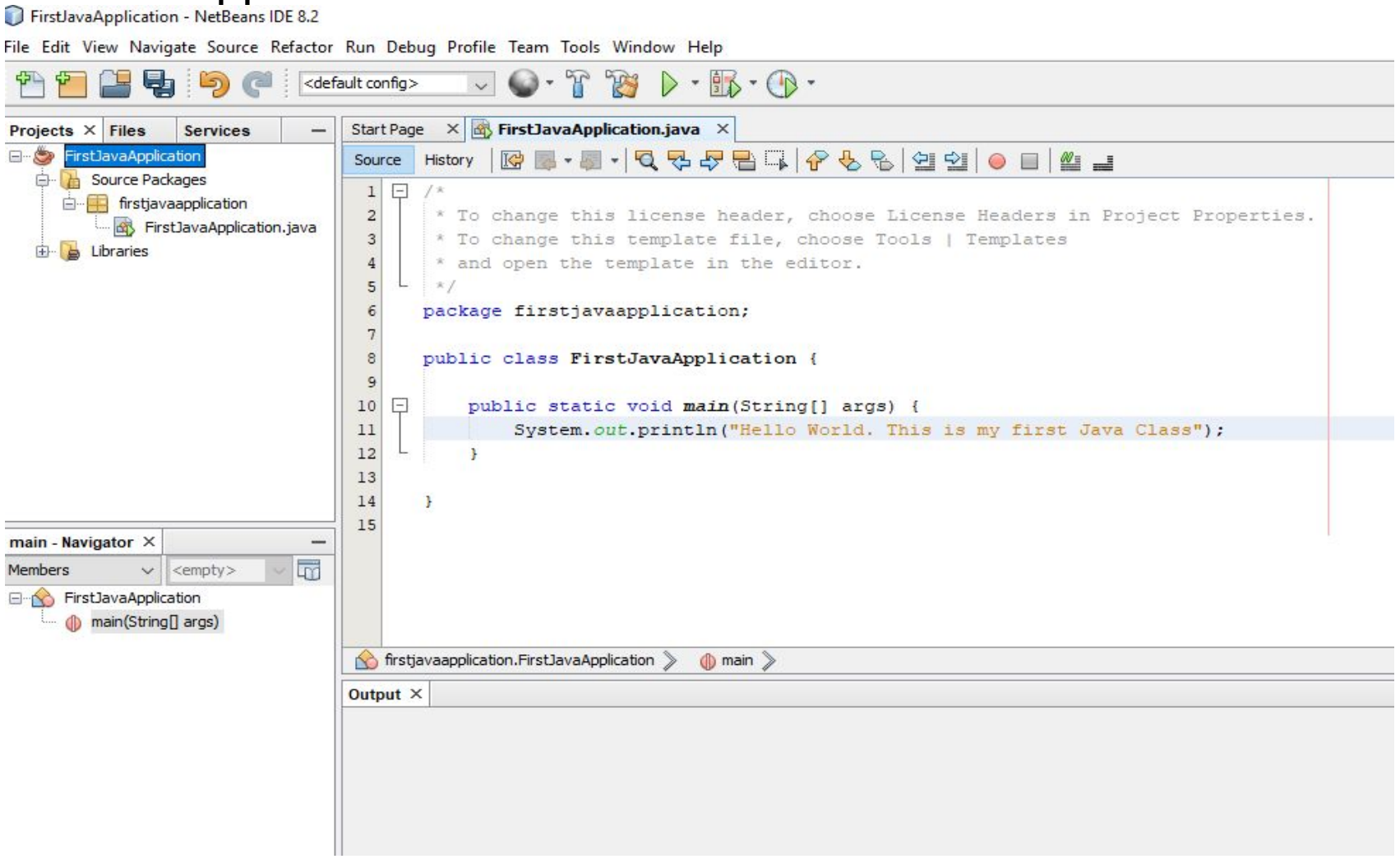
# Creating New Project in NetBeans

❖ Now project with provided project name is created with a project_name.java file created within the project.

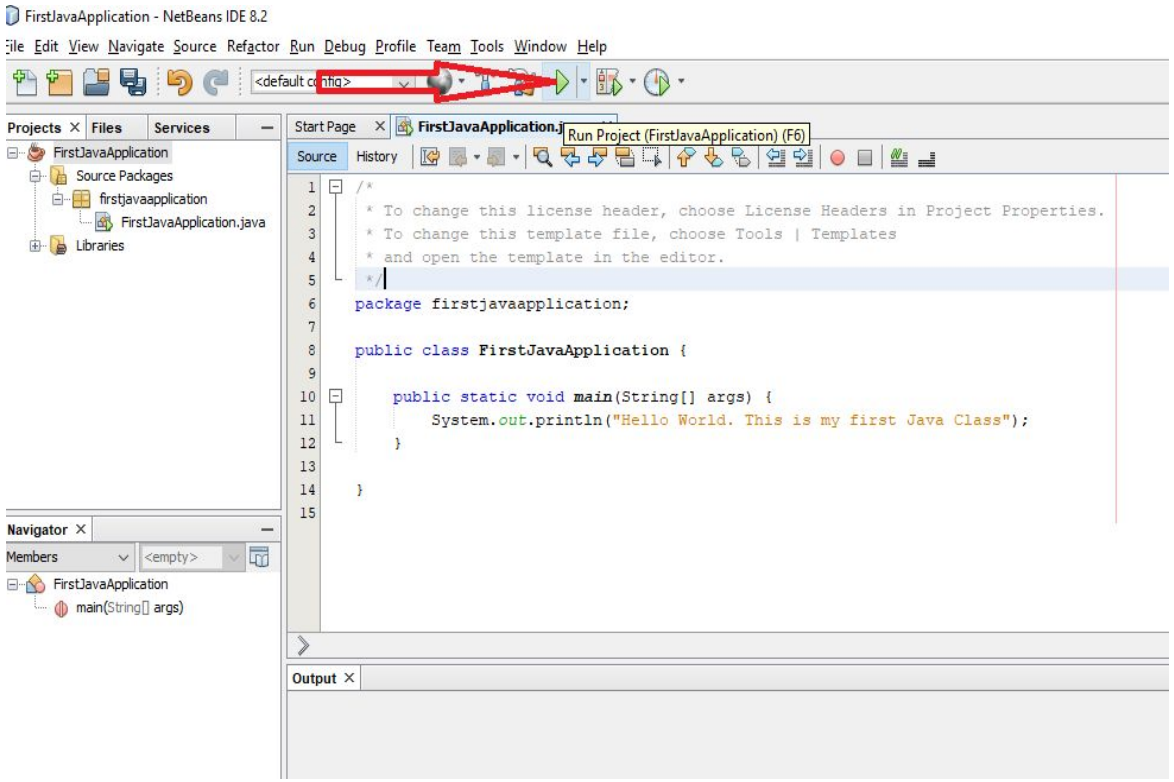


# First Program In Java

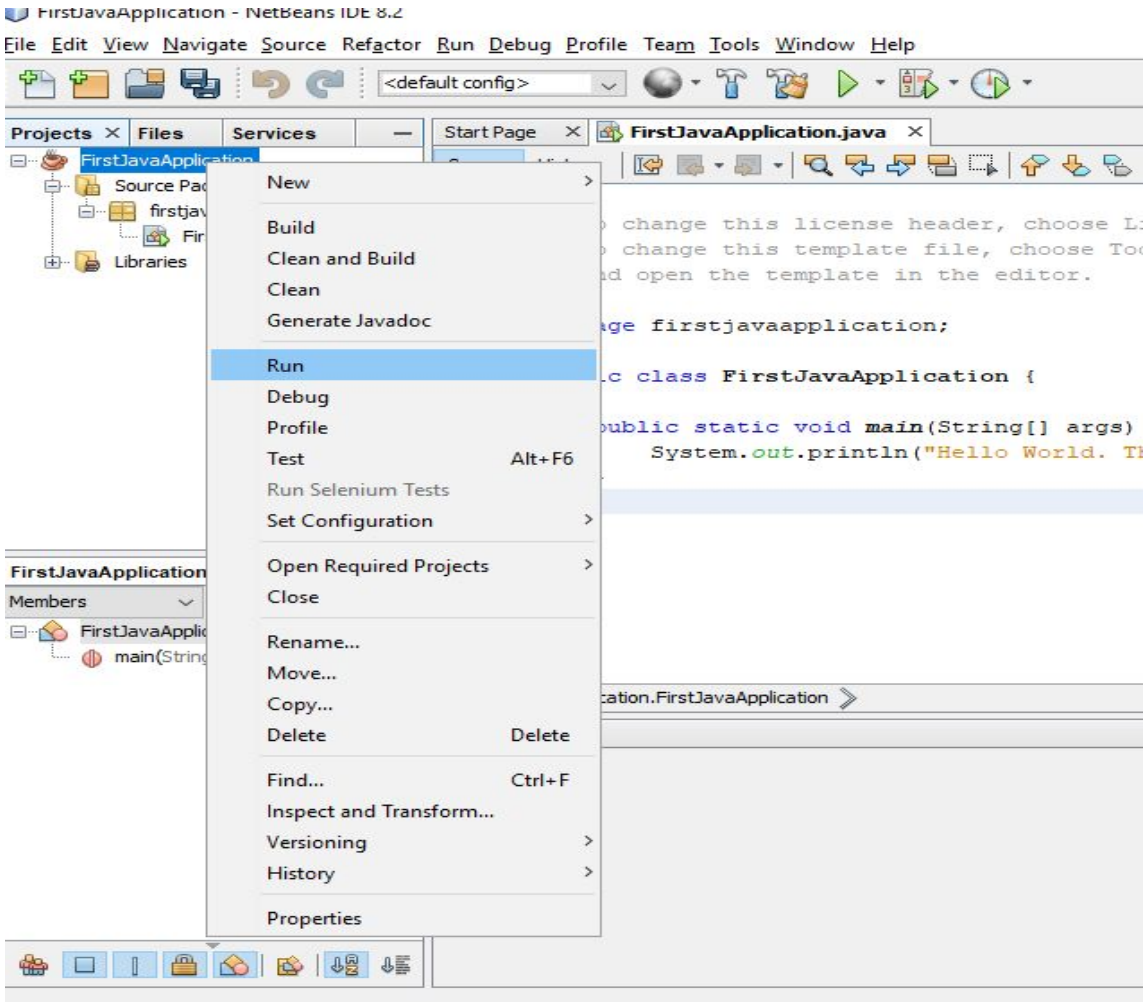
- Suppose we have created a project named FirstJavaApplication. Within that project, a file named FirstJavaApplication will be created within which we write our code.



# Running First Project In Java




❖ Alternatively: Right click on the project and select Run File option



# Running the complete Java Project-Output