

JAVA FUNDAMENTALS

Asmaliza Ahzan
IVERSON ASSOCIATES SDN BHD

Java Fundamentals

The History and Philosophy of Java

- Java was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. in 1991.
- It took 18 months to develop the first working version.
- This language was initially called "Oak," but was renamed "Java" in 1995.
- Based on C++ programming language.

Object-Oriented Programming

- OOP is at the core of Java.
- The 3 OOP principles
 1. **Encapsulation** – the mechanism that binds together code and the data it manipulates and keeps both safe from outside interference and misuse.
 2. **Inheritance** – the process by which one object acquires the properties of another object.
 3. **Polymorphism** – a feature that allows one interface to be used for a general class of actions.

Java Development Kit (JDK)

The Java Development Kit is an implementation of either one of the Java Platform, Standard Edition, Java Platform, Enterprise Edition, or Java Platform, Micro Edition platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, macOS or Windows

Java Runtime Environment (JRE)

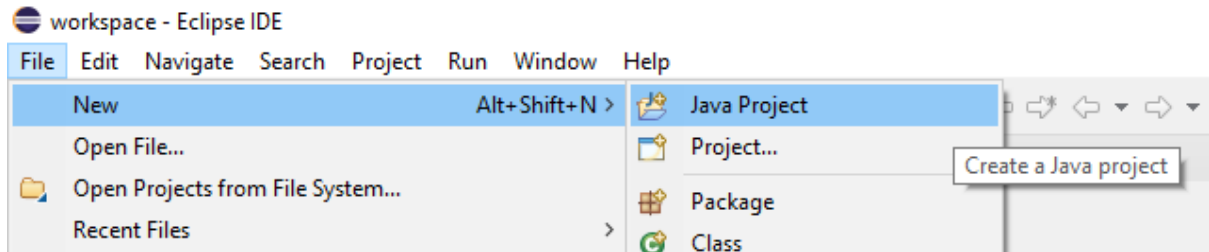
The Java Runtime Environment, or JRE, is a software layer that runs on top of a computer's operating system software and provides the class libraries and other resources that a specific Java program needs to run.

Java Virtual Machine (JVM)

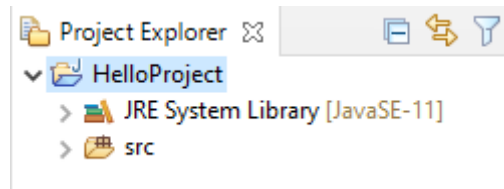
A Java virtual machine (JVM) is a virtual machine that enables a computer to run Java programs as well as programs written in other languages that are also compiled to Java bytecode. The JVM is detailed by a specification that formally describes what is required in a JVM implementation.

A First Simple Program

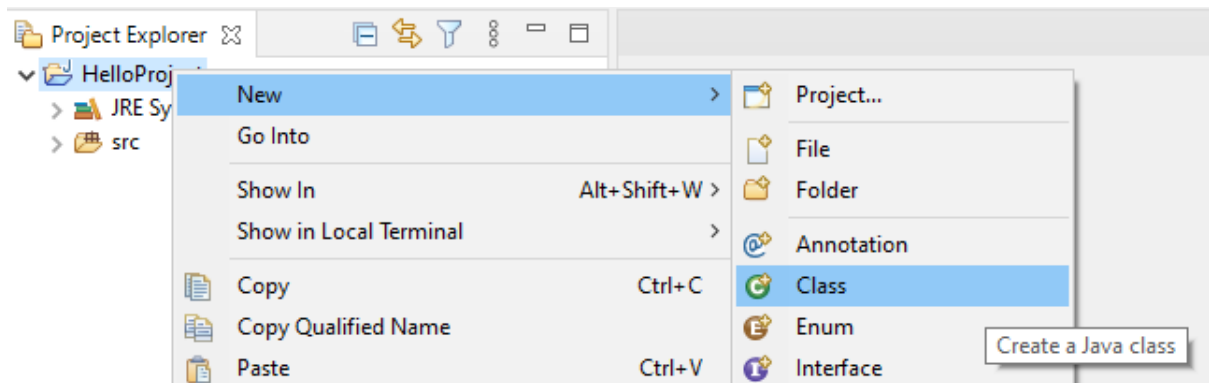
1. Launch eclipse.
2. Close the welcome page.
3. Create a new Java project via File > New > Java Project.



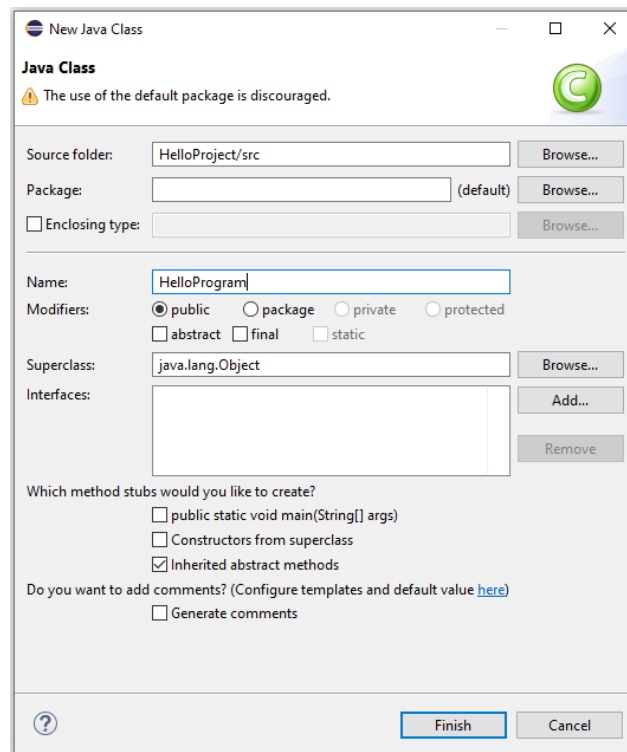
4. In the New Java Project wizard, enter below details.
 - a. Project name: HelloProject
 - b. Module: unchecked
 - c. Leave everything else to default settings and click Finish to proceed.



5. Next is to create a new class. Right-click project > New > Class.



Enter HelloProgram at Name field, then click Finish.



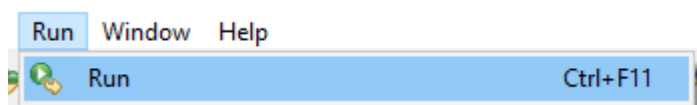
6. Enter below codes in the HelloProgram class.

```
public class HelloProgram {

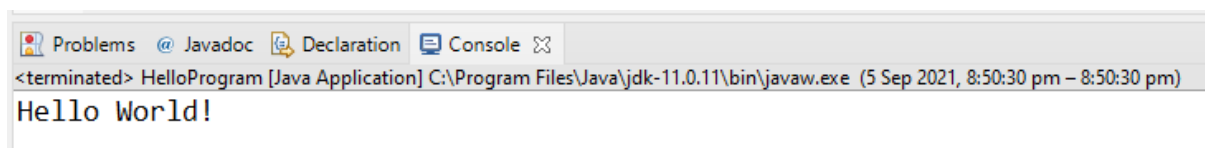
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }

}
```

7. To run the program, go to Run > Run



8. The string "Hello World!" should be displayed in the console/output.



Java Basics

- Java programs are enclosed in a pair of curly brackets that indicates a block of codes.
- Every statement in Java must ends with a semicolon.
- Java is case-sensitive.
- Whitespace in Java program is ignored.

Identifiers

Identifiers are used to name things, such as classes, variables, and methods.

- Must begin with a letter, underscore, or a dollar sign.
- Cannot start with a number.
- A single word.
- case sensitive.
- Cannot use Java keywords.

Comments

- `//` for single-line comments.
- `/* */` for multiple-lines comments.
- `/** */` for Javadoc comments.

The Java Keywords

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

The Java Class Libraries

Full list of Java class libraries can be found at the official page
<https://docs.oracle.com/en/java/javase/11/docs/api/>

All Modules	Java SE	JDK	Other Modules
Module	Description		
java.base	Defines the foundational APIs of the Java SE Platform.		
java.compiler	Defines the Language Model, Annotation Processing, and Java Compiler APIs.		
java.datatransfer	Defines the API for transferring data between and within applications.		
java.desktop	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, imaging, printing, and JavaBeans.		
java.instrument	Defines services that allow agents to instrument programs running on the JVM.		
java.logging	Defines the Java Logging API.		
java.management	Defines the Java Management Extensions (JMX) API.		
java.management.rmi	Defines the RMI connector for the Java Management Extensions (JMX) Remote API.		
java.naming	Defines the Java Naming and Directory Interface (JNDI) API.		