First Tutorial

- link : https://github.com/kos00pas/ECE318_Documentations/blob/main/Tutorial_0/Tutorial_0.md
- First Tutorial
- Educational Licence
- Jetbrains Toolbox
- Download Intellij IDEA Ultimate
- Create a project
  - Setup SDK/JDK
- Setup SDK/JDK for the PC
- Your First Run
- Compile and Intrepreter
- Examples
- Task:
- Further Study:
- Your first shortcuts in Intellij IDEA Ultimate

Educational Licence

- Open the link and fill out the form using your UCY email address
  - https://www.jetbrains.com/shop/eform/students

Jetbrains Toolbox

- Download the application :
  - https://www.jetbrains.com/toolbox-app/

Download Intellij IDEA Ultimate

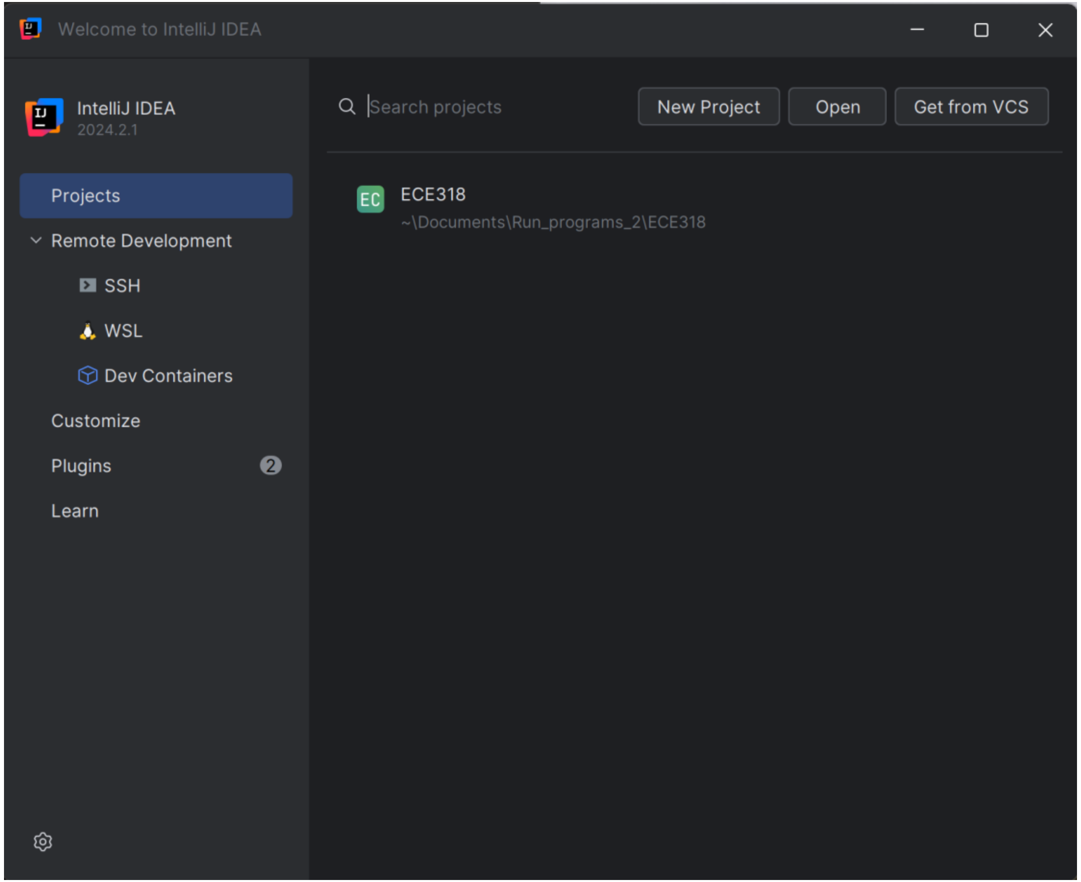
- Open JetBrains Toolbox
- Find Intellij IDEA Ultimate and click Install
- Activate educational licence

Create a project

Open Intellij IDEA Ultimate

Log in with UCY email

In the window `Welcome to Intelij IDEA` window select `New Project`



Setup SDK/JDK

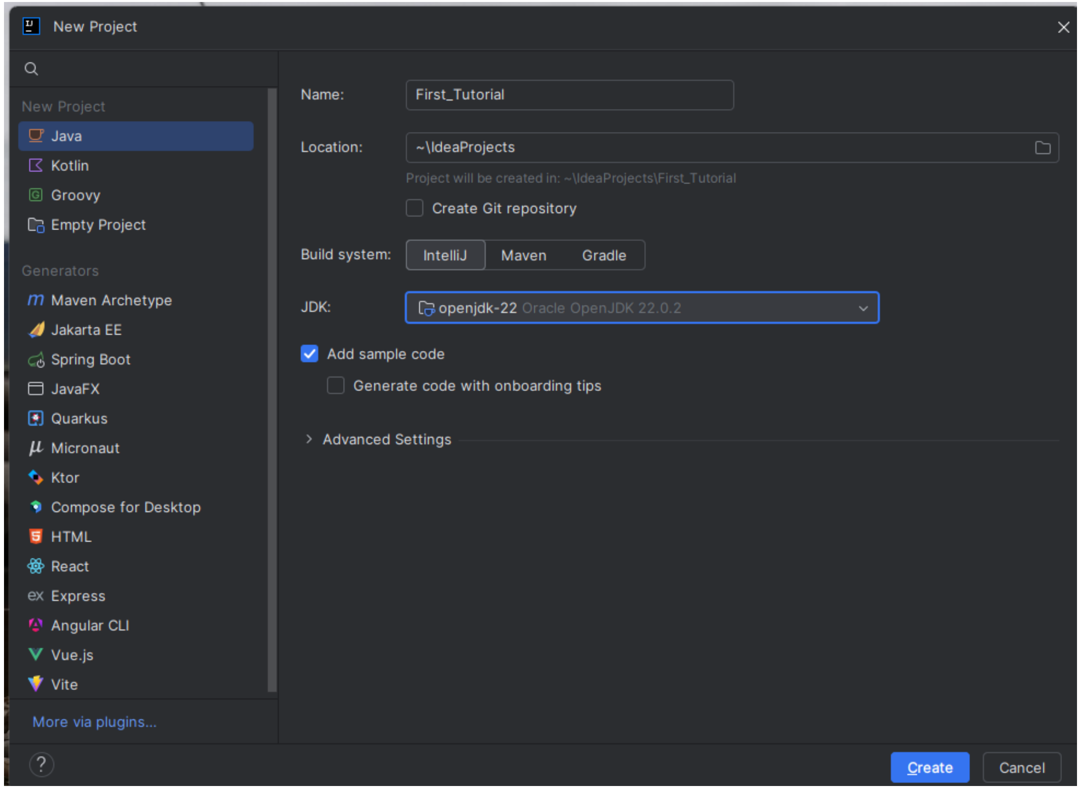
The window below will appear:

Name your project , e.g. “First Tutorial”

Select a location for ECE318 projects and the specific project.

Click on the JDK option and select Download JDK

Note : JDK acts as an SDK within IntelliJ.



New Project

Name: First_Tutorial

Location: ~\IdeaProjects

Project will be created in: ~\IdeaProjects\First_Tutorial

☐ Create Git repository

Build system: IntelliJ | Maven | Gradle

JDK: openjdk-22 Oracle OpenJDK 22.0.2

☑ Add sample code

☐ Generate code with onboarding tips

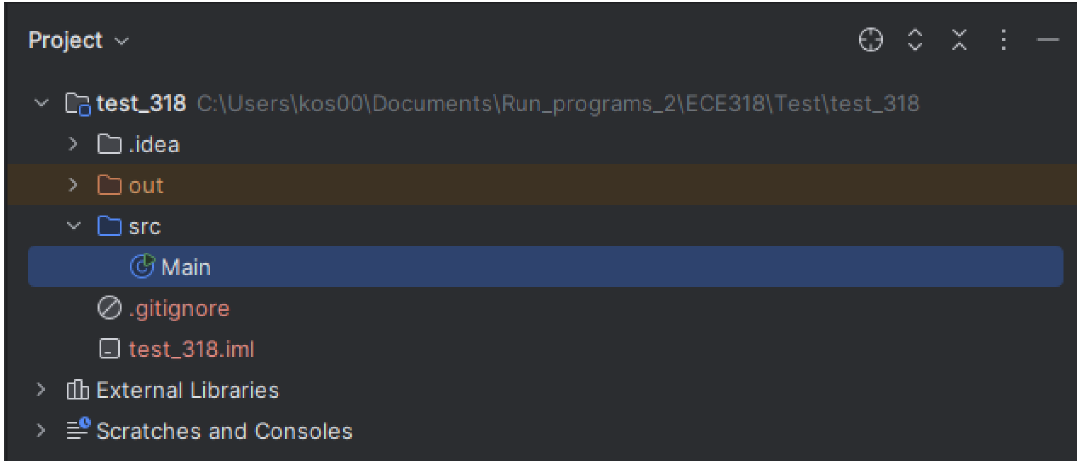
Setup SDK/JDK for the PC

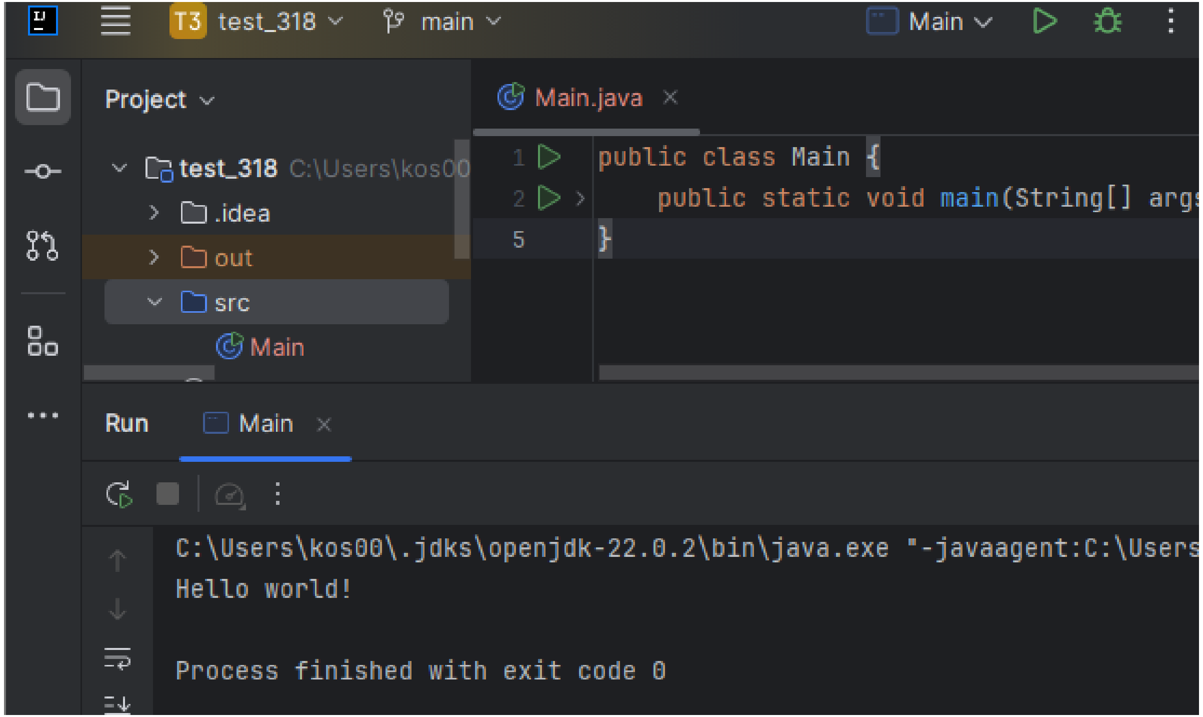
- Open the Environment Variables:
  - Press Windows key + S and type Environment Variables.
  - Click Edit the System Environment Variables.
- Edit the PATH Variable:
  - In the System Properties window, click Environment Variables.
  - Under System variables, locate and select Path, then click Edit.
  - Click New and add this path: `C:\Users\<user_name>\.jdks\openjdk-22.0.2\bin`
  - Click OK to save and close all windows.
- Verify the Setup:
  - Open a new PowerShell or Command Prompt and run: `javac -version`

Your First Run

This is the structure of a project

Click the green button to run the code:

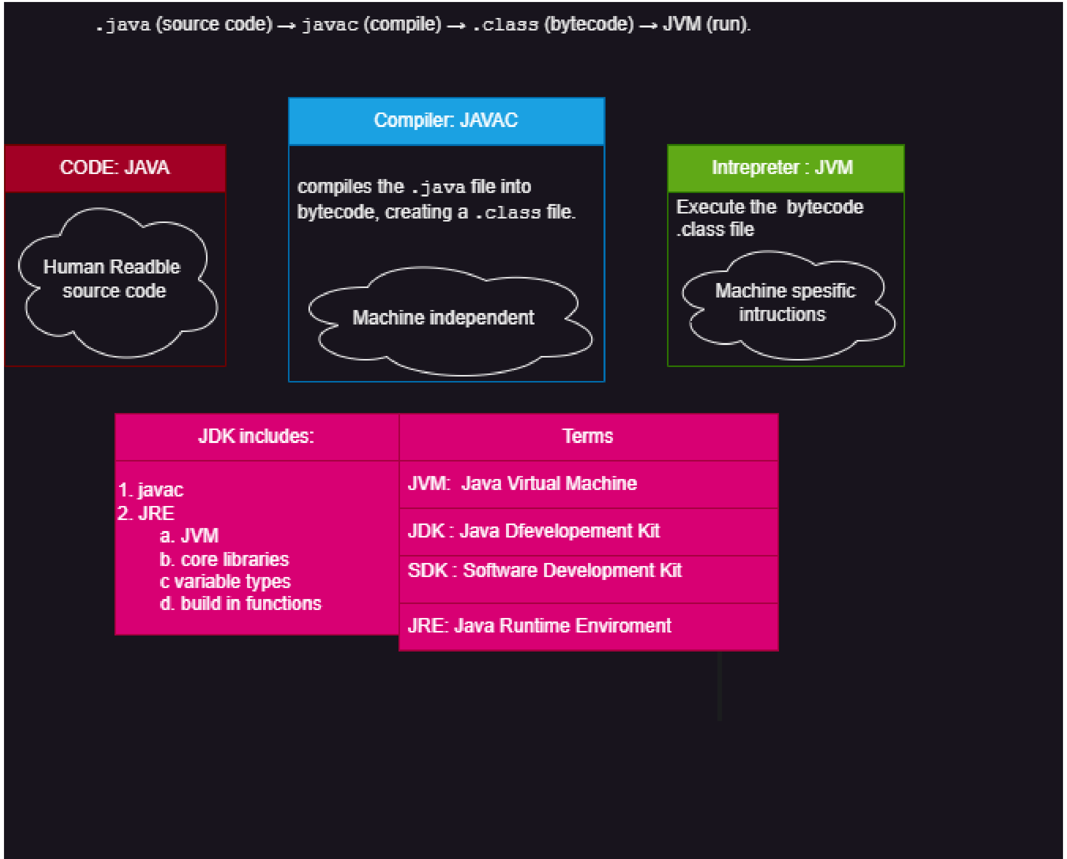




```
public class Main {
    public static void main(String[] args
}
```

```
C:\Users\kos00\.jdks\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Users
Hello world!

Process finished with exit code 0
```

Compile and Intrepreter



`.java` (source code) → `javac` (compile) → `.class` (bytecode) → JVM (run).

CODE: JAVA

Human Readble source code

Compiler: JAVAC

compiles the `.java` file into bytecode, creating a `.class` file.

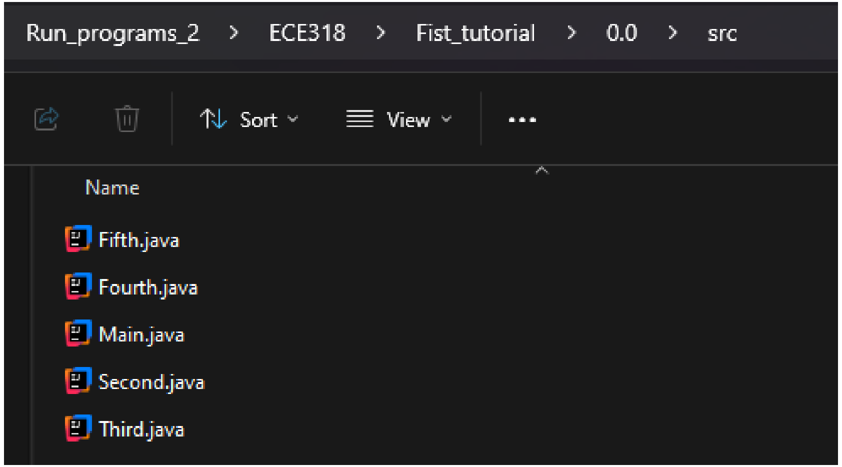
Machine independent

Intrepreter : JVM

Execute the bytecode .class file

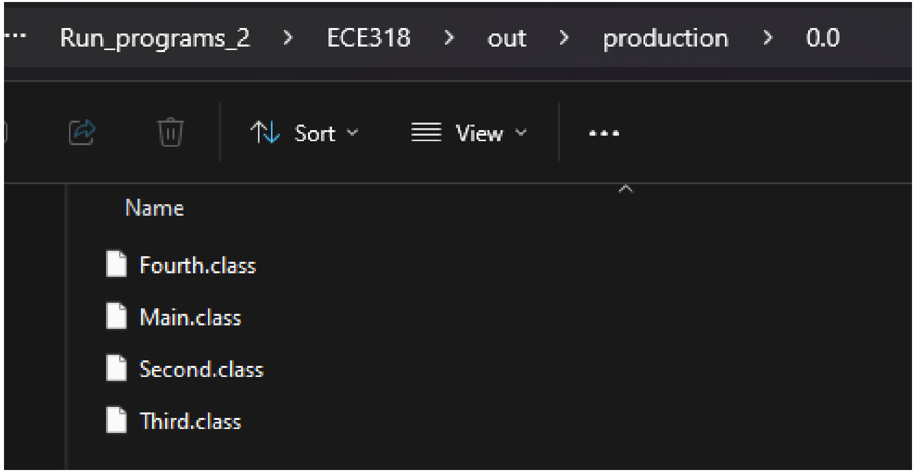
Machine spesific intructions

JDK includes:	Terms
1. javac 2. JRE a. JVM b. core libraries c variable types d. build in functions	JVM: Java Virtual Machine
	JDK : Java Dfevelopement Kit
	SDK : Software Development Kit
	JRE: Java Runtime Enviroment



Run_programs_2 > ECE318 > Fist_tutorial > 0.0 > src

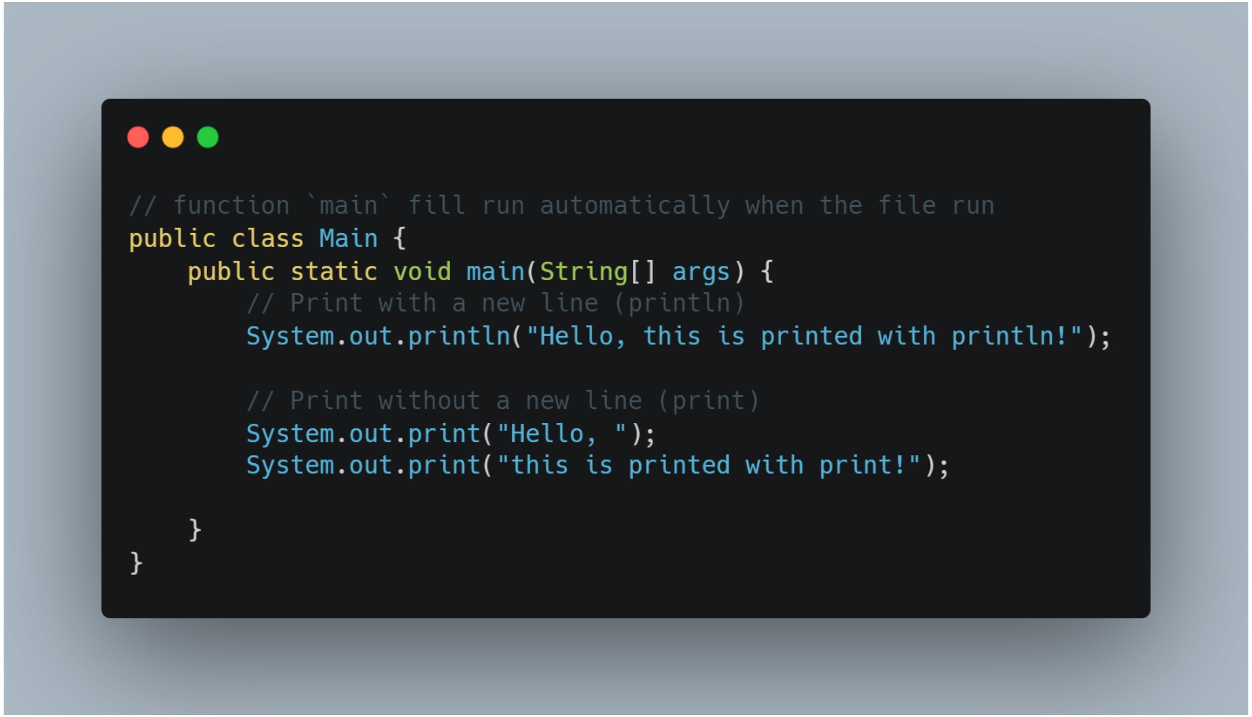
Fifth.java
Fourth.java
Main.java
Second.java
Third.java



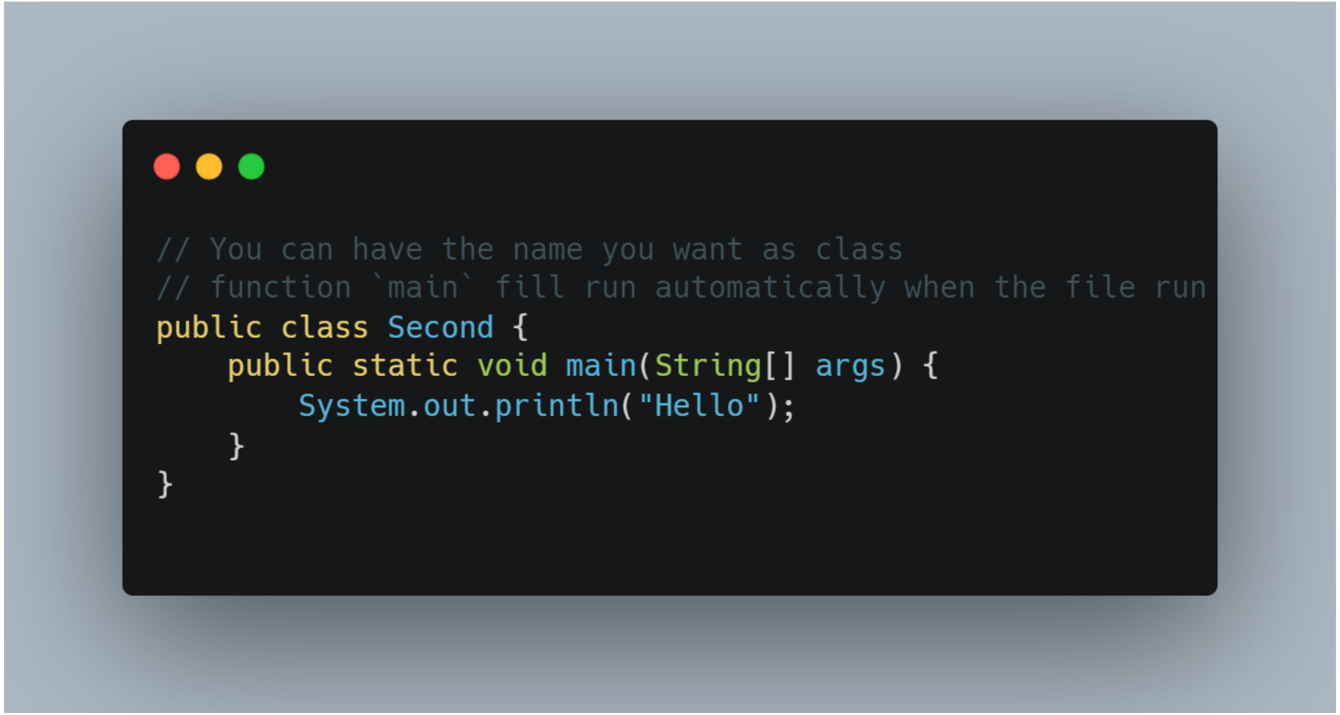
··· Run_programs_2 > ECE318 > out > production > 0.0

Fourth.class
Main.class
Second.class
Third.class

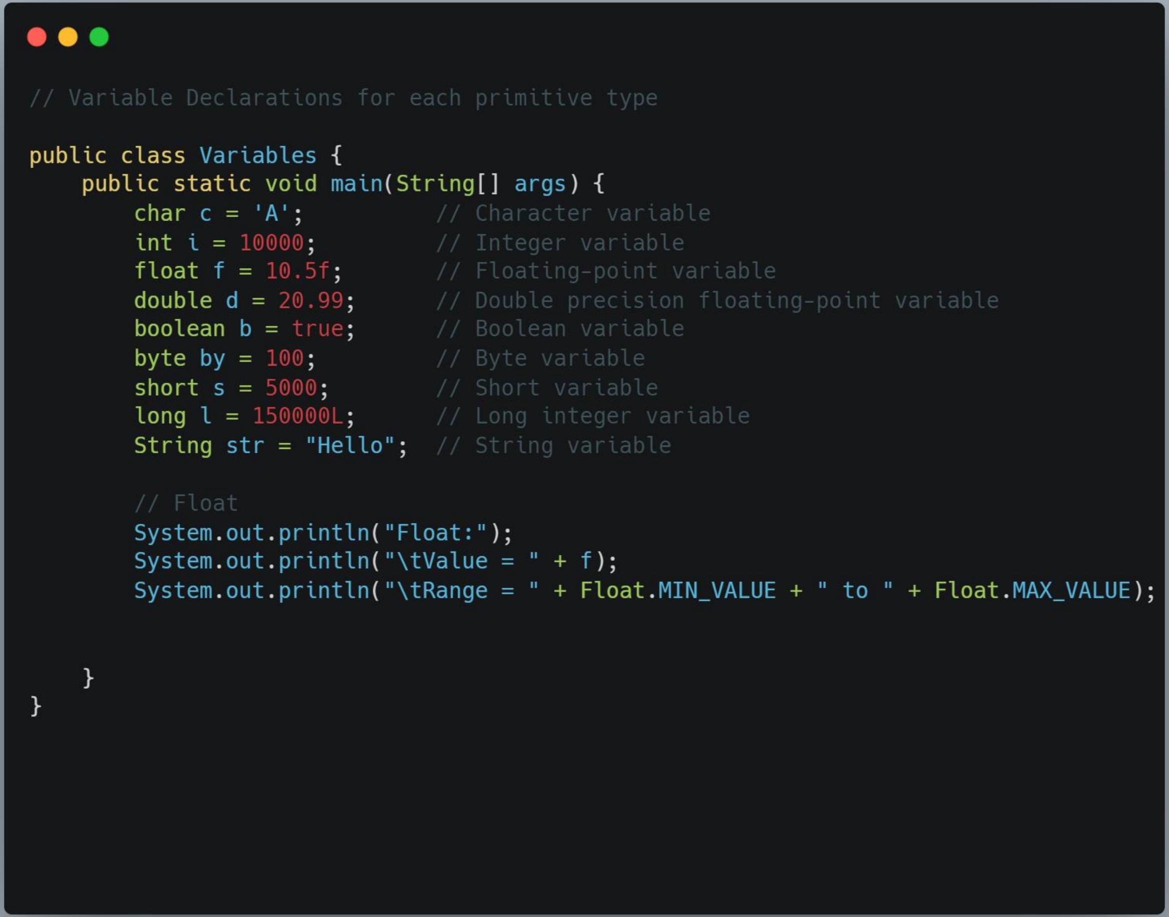
Examples



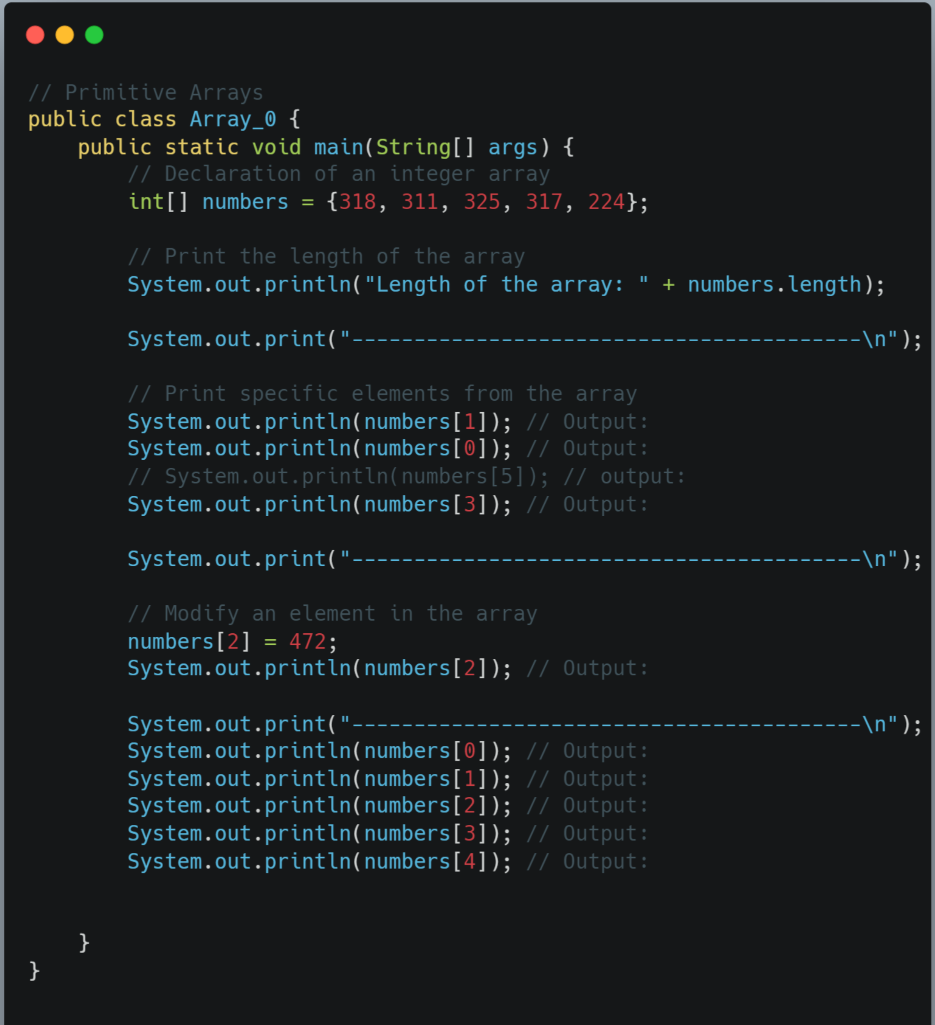
```java
// function `main` fill run automatically when the file run
public class Main {
    public static void main(String[] args) {
        // Print with a new line (println)
        System.out.println("Hello, this is printed with println!");

        // Print without a new line (print)
        System.out.print("Hello, ");
        System.out.print("this is printed with print!");

    }
}
```



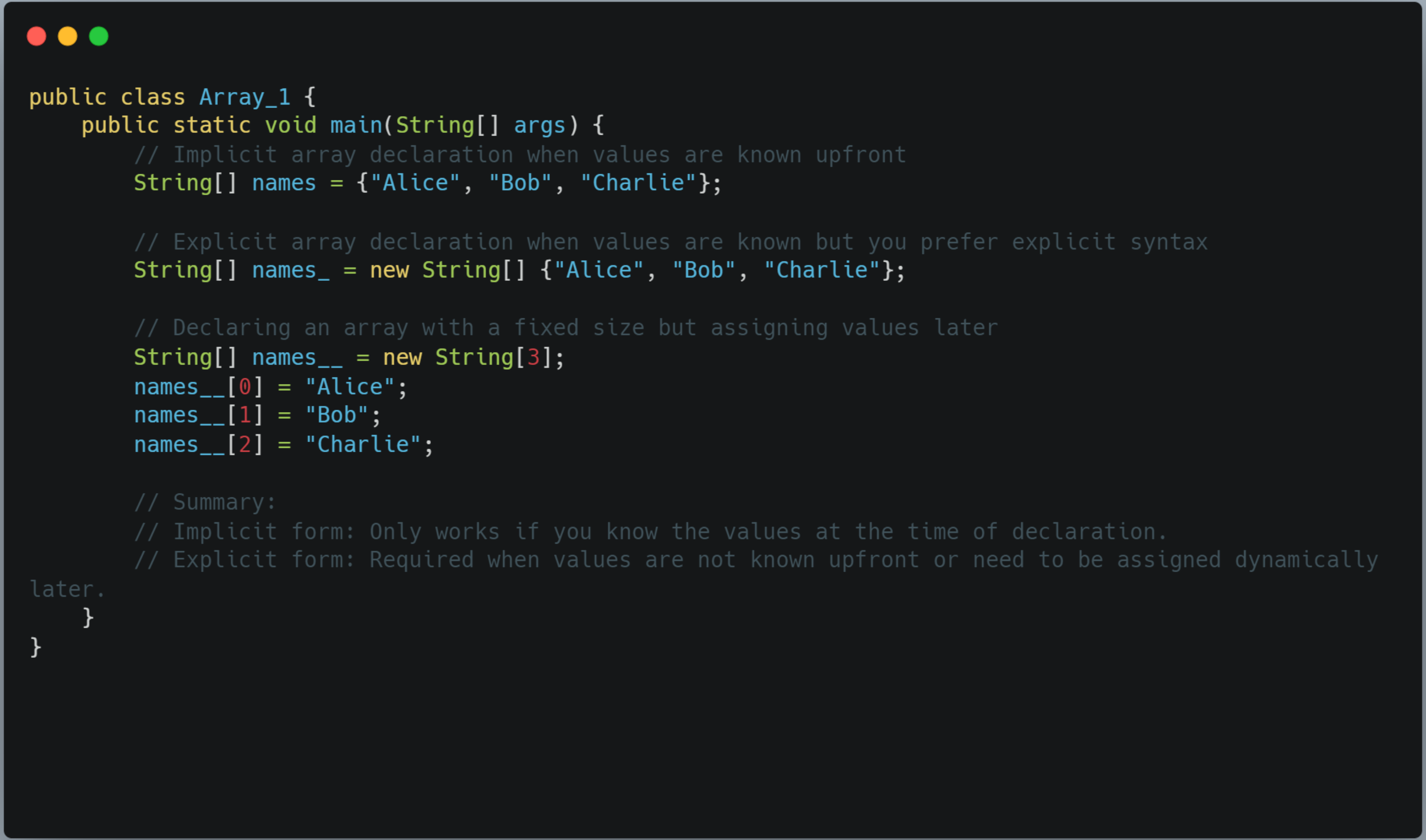
```java
// You can have the name you want as class
// function `main` fill run automatically when the file run
public class Second {
    public static void main(String[] args) {
        System.out.println("Hello");
    }
}
```



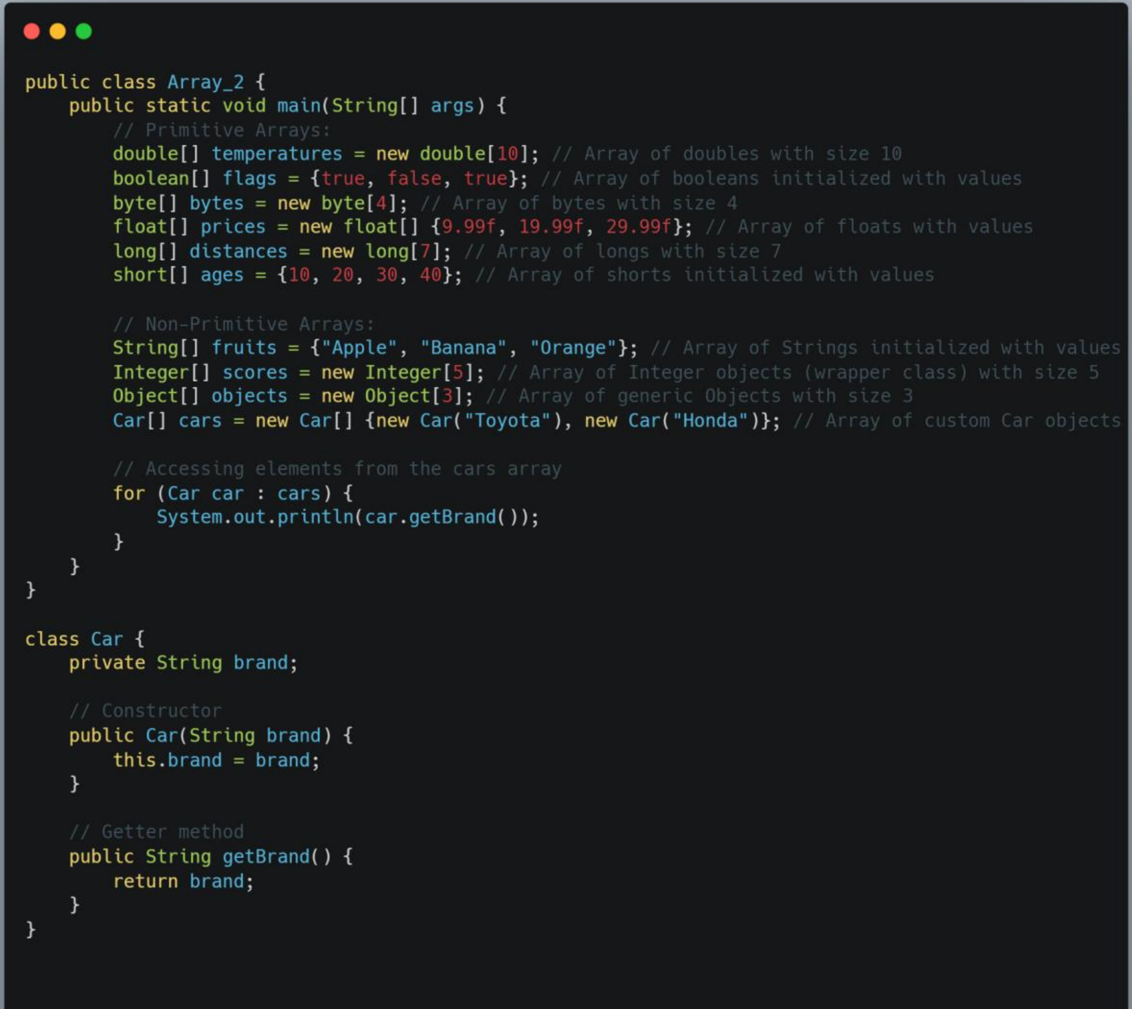
```java
// Variable Declarations for each primitive type

public class Variables {
    public static void main(String[] args) {
        char c = 'A';          // Character variable
        int i = 10000;         // Integer variable
        float f = 10.5f;       // Floating-point variable
        double d = 20.99;      // Double precision floating-point variable
        boolean b = true;      // Boolean variable
        byte by = 100;         // Byte variable
        short s = 5000;        // Short variable
        long l = 150000L;      // Long integer variable
        String str = "Hello";  // String variable

        // Float
        System.out.println("Float:");
        System.out.println("\tValue = " + f);
        System.out.println("\tRange = " + Float.MIN_VALUE + " to " + Float.MAX_VALUE);


    }
}
```



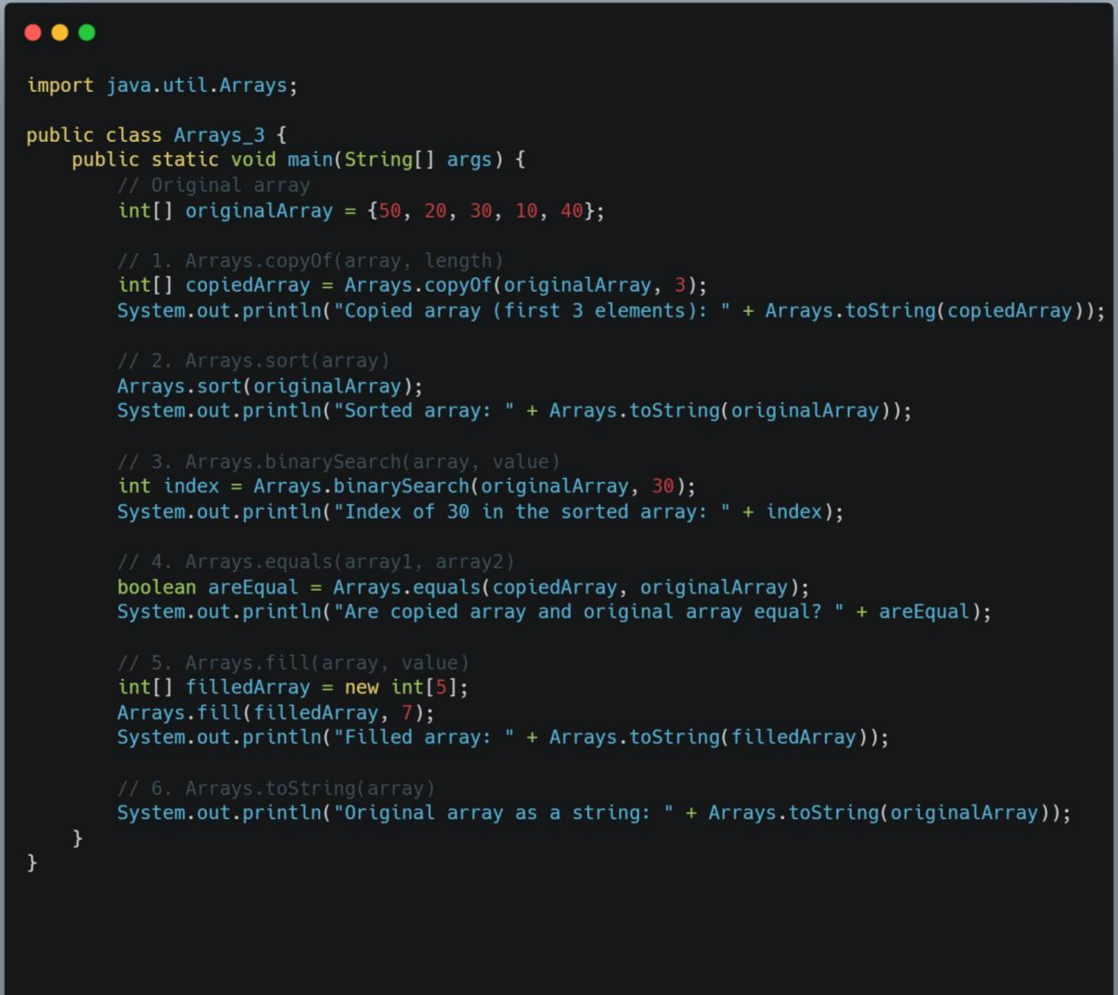
```java
// Primitive Arrays
public class Array_0 {
    public static void main(String[] args) {
        // Declaration of an integer array
        int[] numbers = {318, 311, 325, 317, 224};

        // Print the length of the array
        System.out.println("Length of the array: " + numbers.length);

        System.out.print("-----------------------------------------\n");

        // Print specific elements from the array
        System.out.println(numbers[1]); // Output:
        System.out.println(numbers[0]); // Output:
        // System.out.println(numbers[5]); // output:
        System.out.println(numbers[3]); // Output:

        System.out.print("-----------------------------------------\n");

        // Modify an element in the array
        numbers[2] = 472;
        System.out.println(numbers[2]); // Output:

        System.out.print("-----------------------------------------\n");
        System.out.println(numbers[0]); // Output:
        System.out.println(numbers[1]); // Output:
        System.out.println(numbers[2]); // Output:
        System.out.println(numbers[3]); // Output:
        System.out.println(numbers[4]); // Output:


    }
}
```



```
public class Array_1 {
    public static void main(String[] args) {
        // Implicit array declaration when values are known upfront
        String[] names = {"Alice", "Bob", "Charlie"};

        // Explicit array declaration when values are known but you prefer explicit syntax
        String[] names_ = new String[] {"Alice", "Bob", "Charlie"};

        // Declaring an array with a fixed size but assigning values later
        String[] names__ = new String[3];
        names__[0] = "Alice";
        names__[1] = "Bob";
        names__[2] = "Charlie";

        // Summary:
        // Implicit form: Only works if you know the values at the time of declaration.
        // Explicit form: Required when values are not known upfront or need to be assigned dynamically
later.
    }
}
```



```java
public class Array_2 {
    public static void main(String[] args) {
        // Primitive Arrays:
        double[] temperatures = new double[10]; // Array of doubles with size 10
        boolean[] flags = {true, false, true}; // Array of booleans initialized with values
        byte[] bytes = new byte[4]; // Array of bytes with size 4
        float[] prices = new float[] {9.99f, 19.99f, 29.99f}; // Array of floats with values
        long[] distances = new long[7]; // Array of longs with size 7
        short[] ages = {10, 20, 30, 40}; // Array of shorts initialized with values

        // Non-Primitive Arrays:
        String[] fruits = {"Apple", "Banana", "Orange"}; // Array of Strings initialized with values
        Integer[] scores = new Integer[5]; // Array of Integer objects (wrapper class) with size 5
        Object[] objects = new Object[3]; // Array of generic Objects with size 3
        Car[] cars = new Car[] {new Car("Toyota"), new Car("Honda")}; // Array of custom Car objects

        // Accessing elements from the cars array
        for (Car car : cars) {
            System.out.println(car.getBrand());
        }
    }
}

class Car {
    private String brand;

    // Constructor
    public Car(String brand) {
        this.brand = brand;
    }

    // Getter method
    public String getBrand() {
        return brand;
    }
}
```



```java
import java.util.Arrays;

public class Arrays_3 {
    public static void main(String[] args) {
        // Original array
        int[] originalArray = {50, 20, 30, 10, 40};

        // 1. Arrays.copyOf(array, length)
        int[] copiedArray = Arrays.copyOf(originalArray, 3);
        System.out.println("Copied array (first 3 elements): " + Arrays.toString(copiedArray));

        // 2. Arrays.sort(array)
        Arrays.sort(originalArray);
        System.out.println("Sorted array: " + Arrays.toString(originalArray));

        // 3. Arrays.binarySearch(array, value)
        int index = Arrays.binarySearch(originalArray, 30);
        System.out.println("Index of 30 in the sorted array: " + index);

        // 4. Arrays.equals(array1, array2)
        boolean areEqual = Arrays.equals(copiedArray, originalArray);
        System.out.println("Are copied array and original array equal? " + areEqual);

        // 5. Arrays.fill(array, value)
        int[] filledArray = new int[5];
        Arrays.fill(filledArray, 7);
        System.out.println("Filled array: " + Arrays.toString(filledArray));

        // 6. Arrays.toString(array)
        System.out.println("Original array as a string: " + Arrays.toString(originalArray));
    }
}
```

Task:

1. **Run** the above and interact with the outputs.
2. **Create a program** with the name `MyRecord.java` that will print your information.
3. The program should **print out** the following information using `System.out.println();`:

- **Name and Surname** (e.g., Angelos Marnerides)
- **Date of birth** (e.g., 01/05/1980)
- **Town** (e.g., Larnaca)
- **Sex** (e.g., M or F)
- **Course codes taken** for this semester (e.g., ECE318, ECE311)
- You should store the corresponding information in arrays and then printing the array(s).

Further Study:

- variables : - https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html
- Arrays :
  - https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html
  - https://www.geeksforgeeks.org/array-class-in-java/?ref=lbp
- java.util:
  - https://docs.oracle.com/javase/8/docs/api/java/util/package-summary.html

Your first shortcuts in Intellij IDEA Ultimate

- Set up mouse wheel for zoom in/out:
  1. File -> project structure
     - Go to File > Settings -> Editor > General.
       - Find the section : `Mouse Control`.
       - Check the box `Change font size (Zoom) with Ctrl+Mouse Wheel`.
       - Click Apply and then OK.
- Comments
  - `//`
  - Ctrl + `/` -> comment a line or multiple
  - Ctrl + Shift + `/` -> comment with `*/`
- Add ';' in the end of a line:
  - Ctrl+ Shift + Enter
- Close and open `Project Window`
  - Open : Alt + 1
  - Close: Shift + Esc
- Run code
  - Ctrl+Shift+ f10 -> run current code
  - Fn+ Shift + f10 -> Run the last run code