

Welcome to the  
**Java**  
**Course**

Module 1 – Day 1

# Contents of the module

- **Introduction to programming**
- **Basic programming concepts**
- **Variables and Data Types**
- Conditionals
- Loops
- Control Structures
- Introduction to algorithms

# Introduce yourself

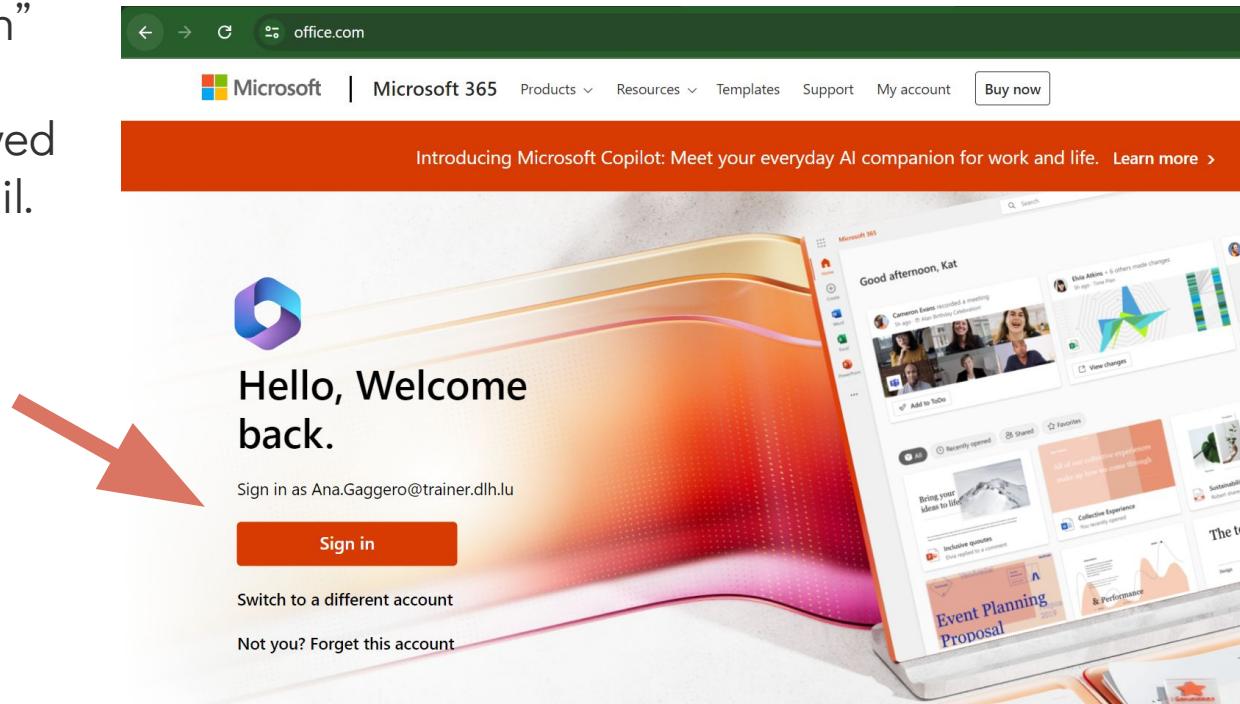
1. Name
2. Background
3. What brings you to Java Module 1?
4. What do you expect from this course?

# Log into the local PC and to Microsoft Teams

Open Google Chrome and type: office.com

Step 1: Click on “Sign in”

You should have received  
your password by email.



# Log into the local PC and to Microsoft Teams

Step 2: Click on “Teams”

1



explore by category

Productivity

Utilities

Education

Communication

Content management

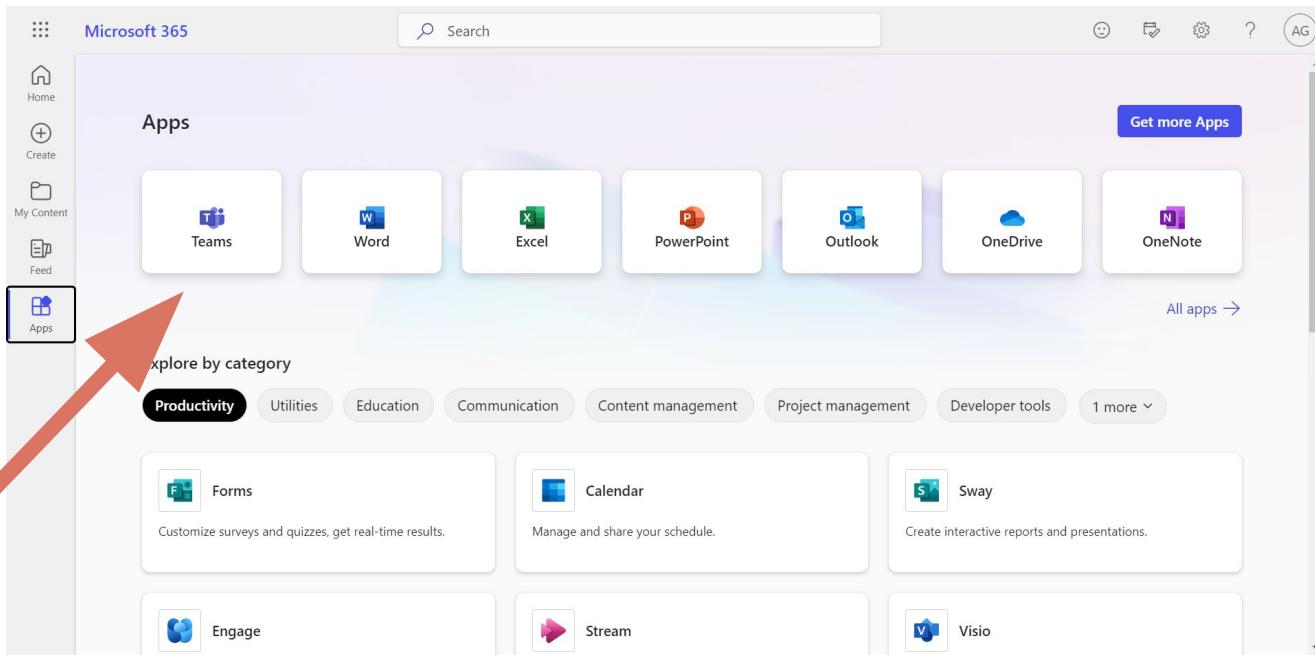
Project management

Developer tools

1 more

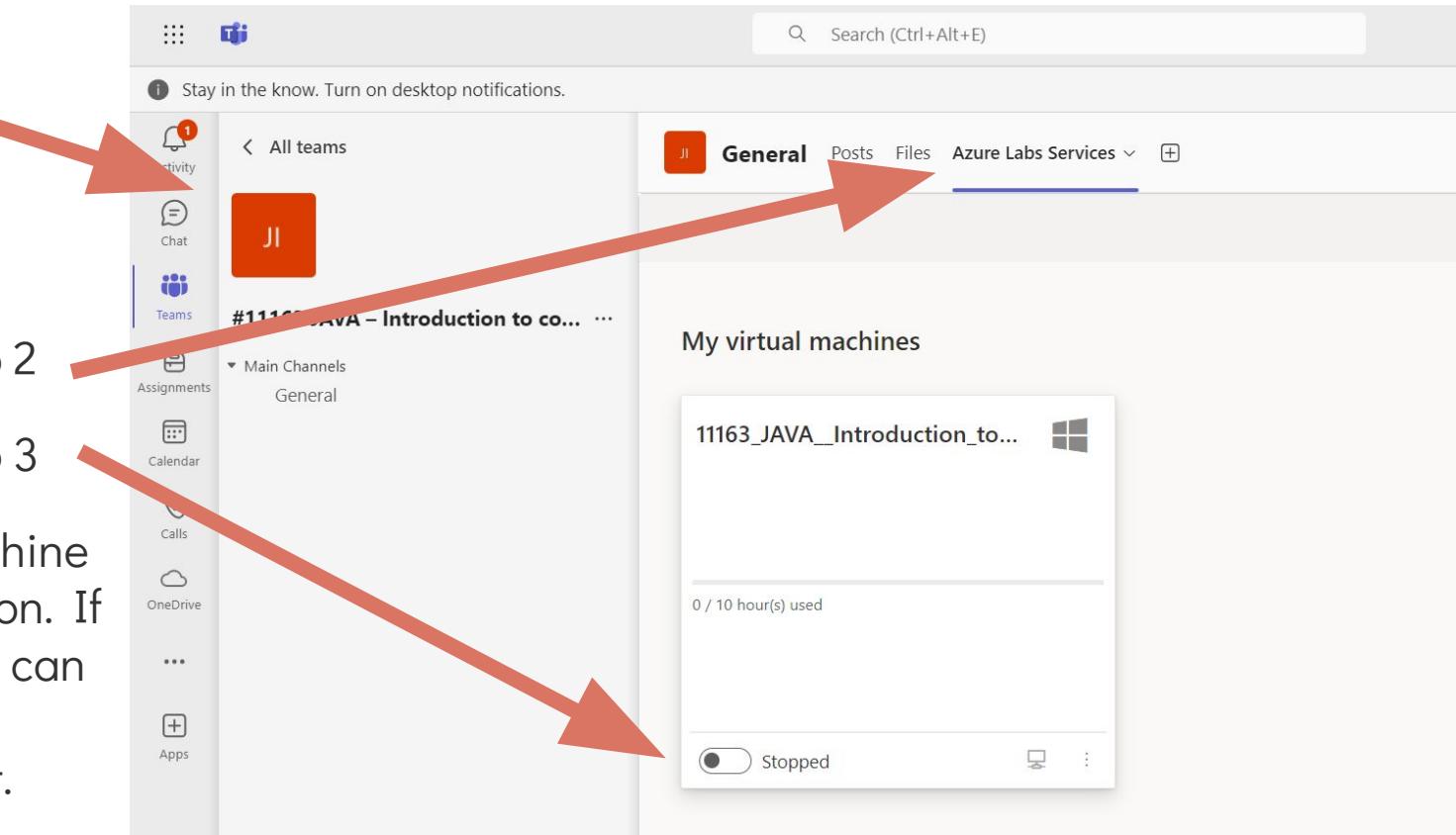
All apps →

2



# Log into the virtual machine

Step 1



The virtual machine will be already on. If not, turn it on, it can take up to 5 minutes to start.

# Log into the virtual machine

**Username:** student

**Password:** StudentDLH2024

# Project Students

Building a program that **gathers information** from students and **validates** it.

- 1. Student Registration**
- 2. Data Validation**

# Project Tic Tac Toe

A program that allows two players to play Tic Tac Toe against each other

```
What is your name Player 1? Ana  
What is your name Player 2? Juan
```

```
Ana will be X and Juan will be O
```

1		2		3
4		5		6
7		8		9

```
Ana choose your next move: 4
```

1		2		3
X		5		6
7		8		9

# Computational Thinking

Problem-solving approach  
that helps you to think  
**logically** and solve problems  
**effectively.**

# Computational Thinking

## **Decomposition:**

Breaking down a complex problem  
into smaller, more manageable  
sub-problems or tasks.

# Computational Thinking

## **Pattern Recognition:**

Identifying recurring patterns or similarities in different problems to develop generic solutions.

# Computational Thinking

## **Abstraction:**

Focusing on essential details while ignoring irrelevant information to simplify the problem.

# Computational Thinking

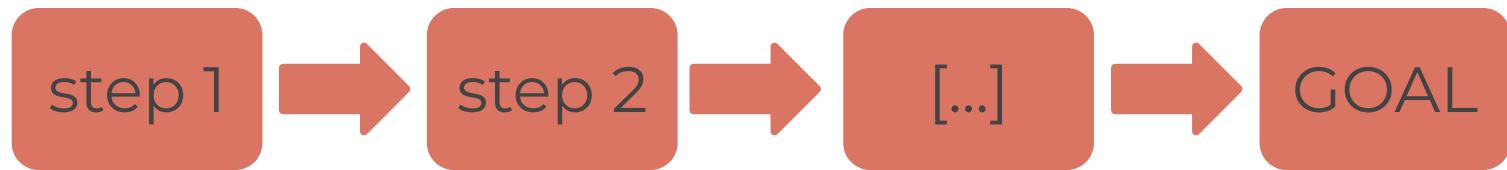
## **Evaluation and Optimization:**

Evaluate the efficiency and effectiveness of solutions and refining them for better performance.

# Algorithm

An Algorithm is an ordered **series of steps** that leads to the achievement of a goal or the solution to a problem.

# Algorithm



# Computational Thinking

It is not limited to computer science but can be applied to various disciplines and **real-world challenges.**

# Computational Thinking

Make your sandwich

Decompose  
step by step



# Computational Thinking

- 1. Prepare the ingredients**
- 2. Assemble the ingredients**
- 3. Plate it**



*Concept: Decomposition and Abstraction*

# Computational Thinking

1. Prepare the ingredients

1. **Cut the slices of bread**

2. **Cut tomato**

3. **Take the ham out of the fridge**

4. **Take the salad out of the fridge**



# Computational Thinking

1. Prepare the ingredients
2. Assemble the ingredients
  1. **Slice of bread**
  2. **Salad**
  3. **Ham**
  4. **Tomato slices**
  5. **Slice of bread**



# Computational Thinking

1. Prepare the ingredients
2. Assemble the ingredients
3. Plate it

**1. Put it on a plate**



*Concept: Algorithm*

# Computational Thinking

Now, make a Lasagna

1. Prepare the ingredients
2. Assemble the ingredients
3. ~~Plate it~~ Cook it in the oven



Concept: Pattern

# Programming Language

is a language that humans can use and that the machine can understand.

# Programming Language



Swift



# Java

One of the **most used** programming language. For example, it's used in the development of **desktop** and **mobile** applications.



Adobe



Spotify®

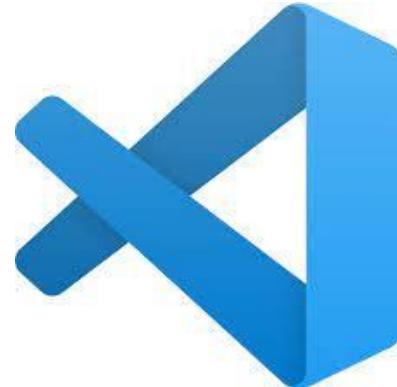
MINECRAFT



# Code Editor

To develop in a programming language, you need a code editor that understands and executes the language. In this course we will mainly use a very common one:

Visual Studio Code



# Output

```
System.out.println("text");
```

Let's create your first

**Hello world !**

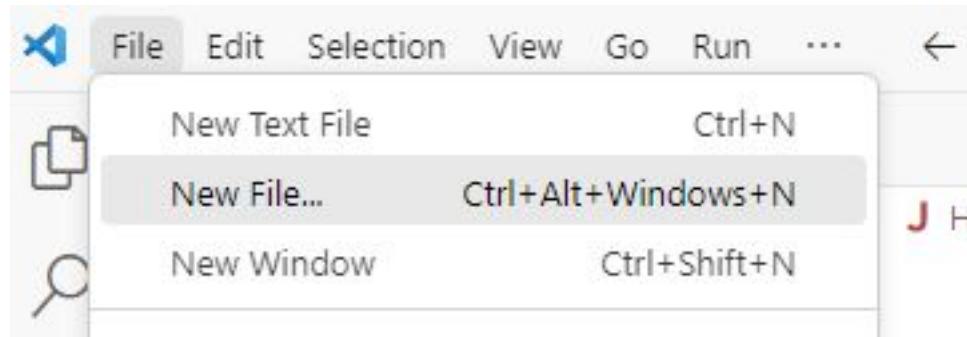
# Code Editor

Create a new program



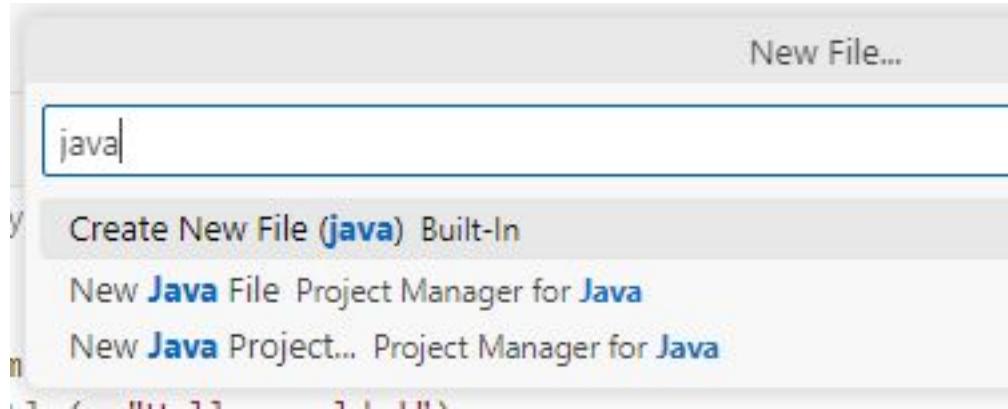
# Code Editor

Step 1: **Create** a file.



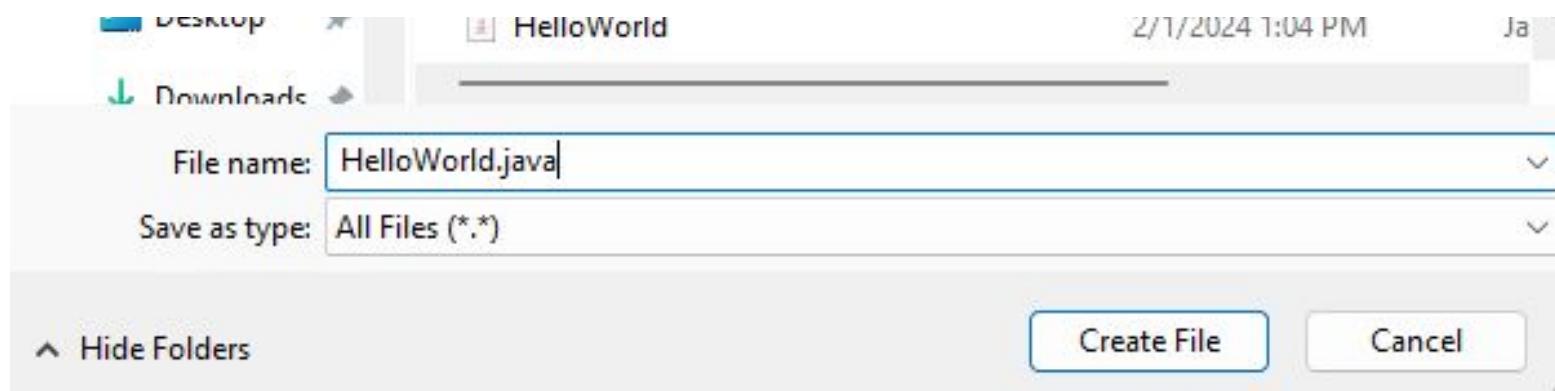
# Code Editor

Step 2: Create New File (**java**) Build-in



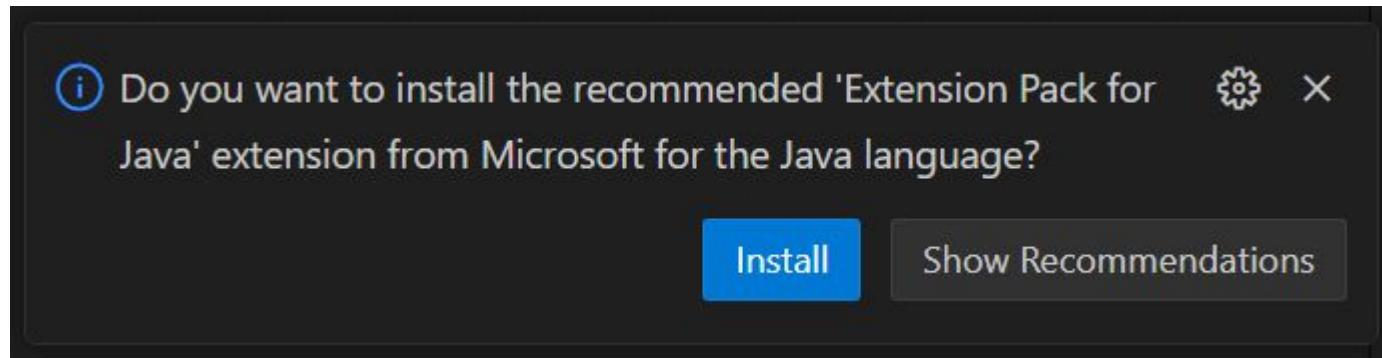
# Code Editor

Step 4: **Name** of your class (**Capital** letter and **no space** in the name)



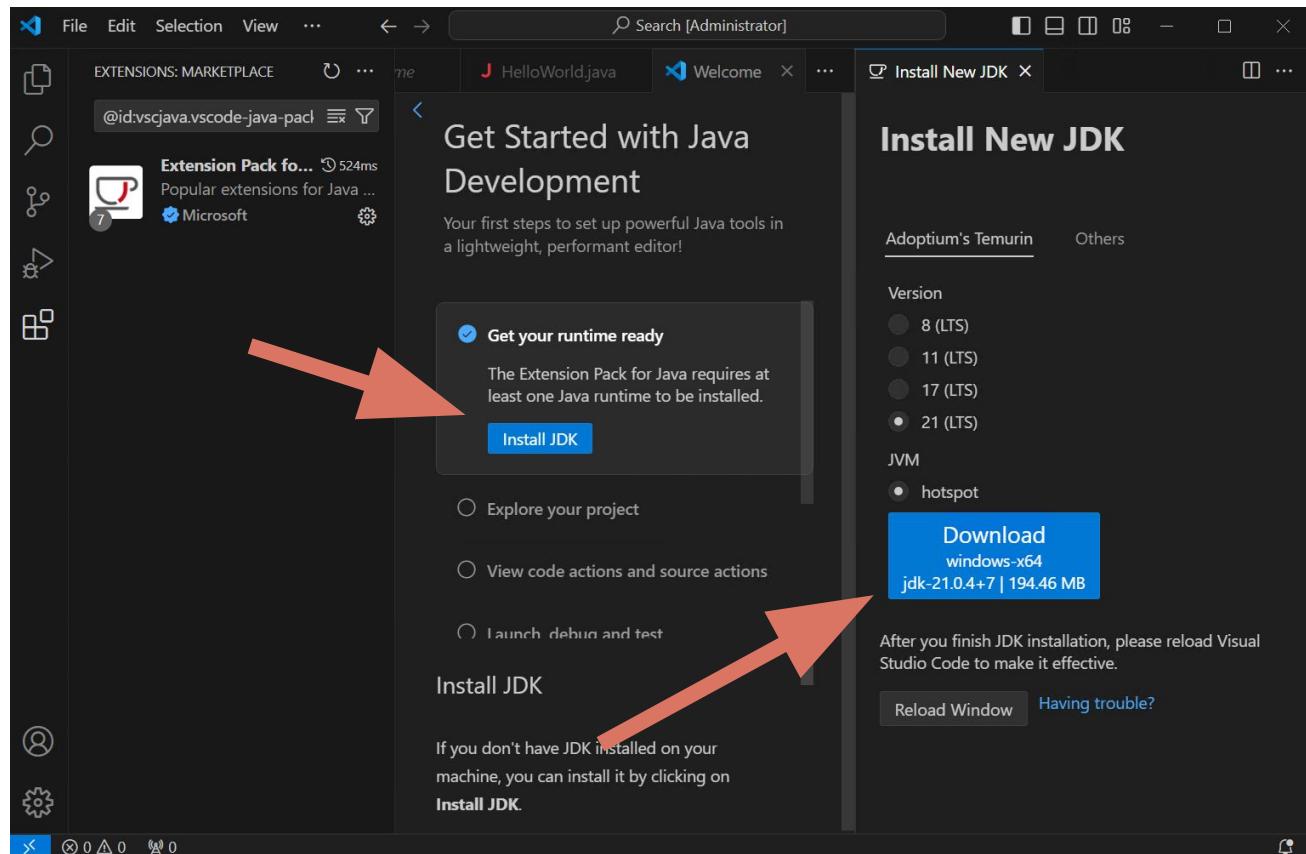
# Code Editor

Step 5: When you create a .java file for the first time, the editor will offer to install the Extension pack for Java.



# Code Editor

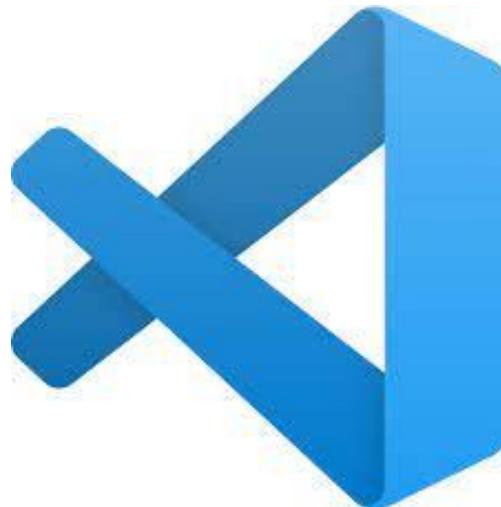
Step 6: When it finishes installing the extension, the editor will ask you to install the Java Development Kit (JDK)



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

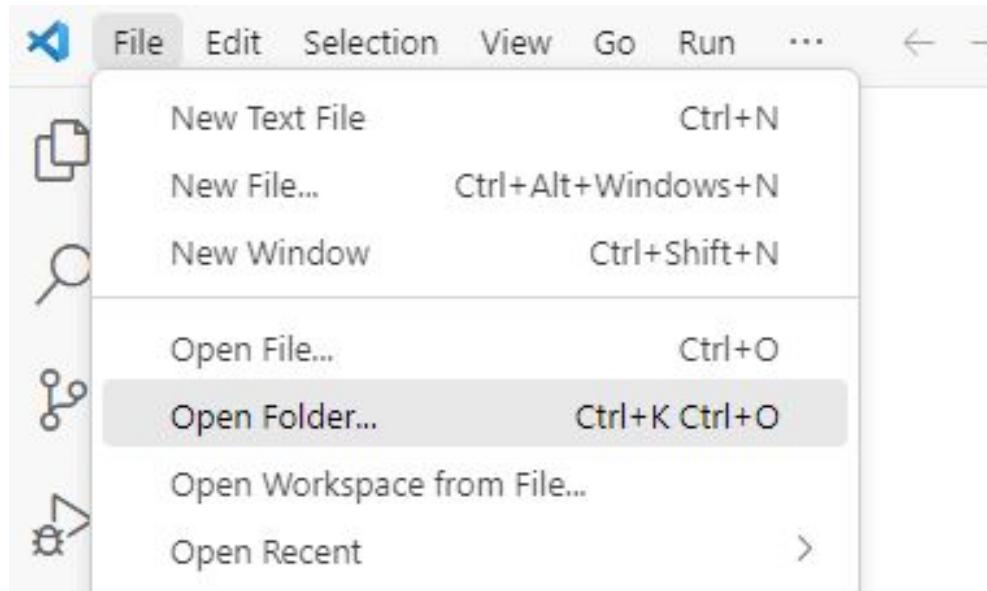
# Code Editor

Open and run an existing program



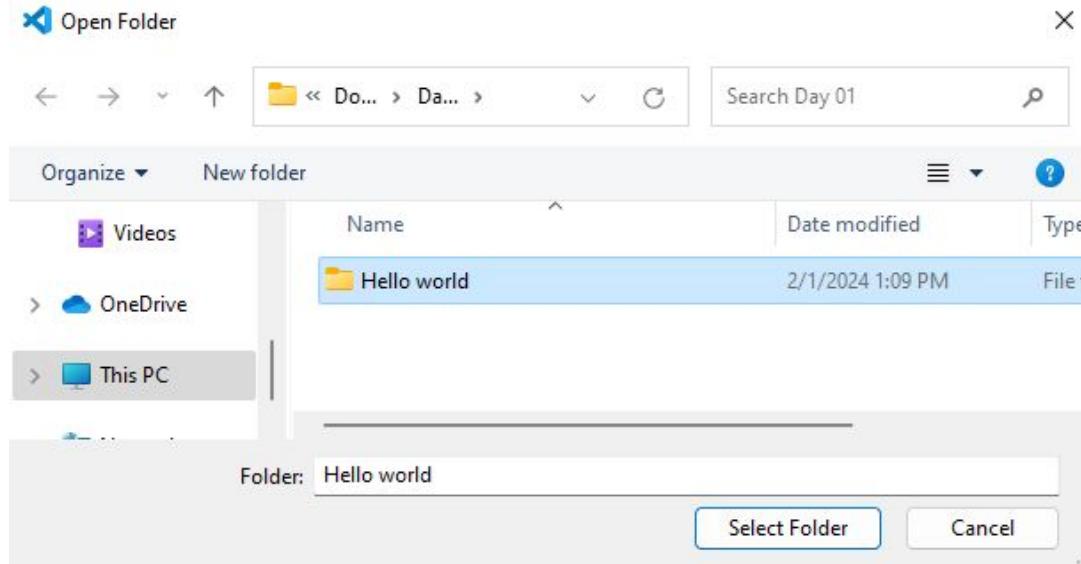
# Code Editor

Step 1: **Open** the main folder.

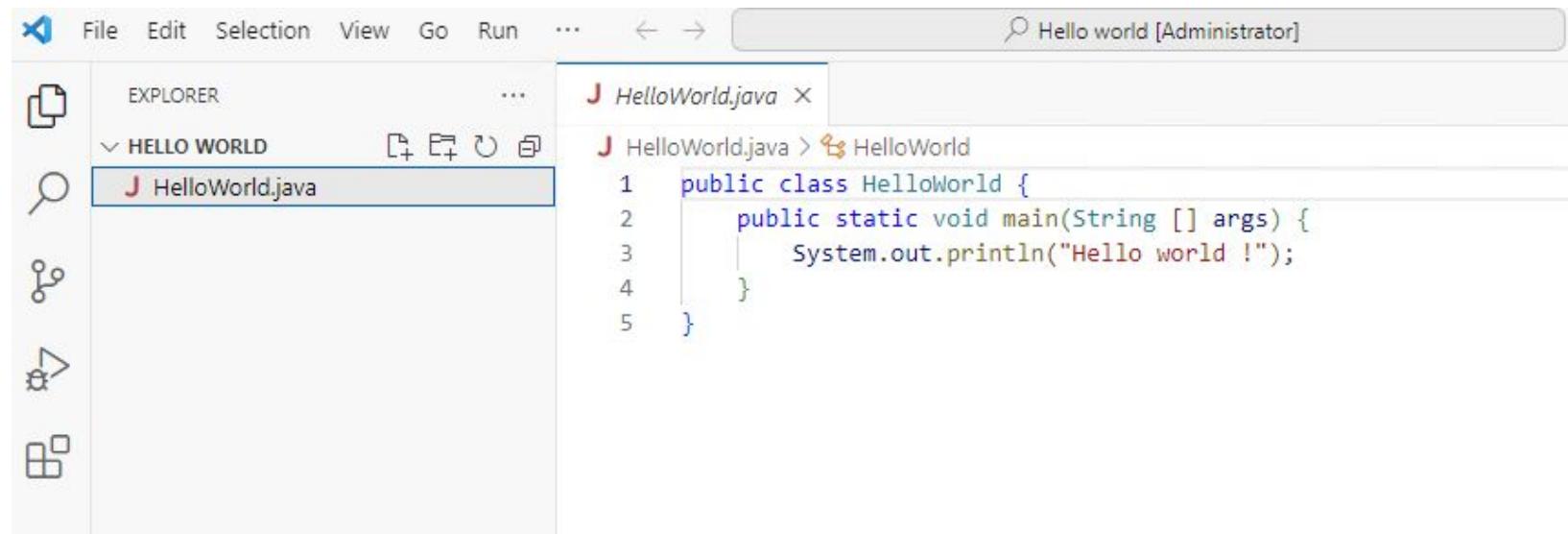


# Code Editor

Step 2: **Choose** the main folder.

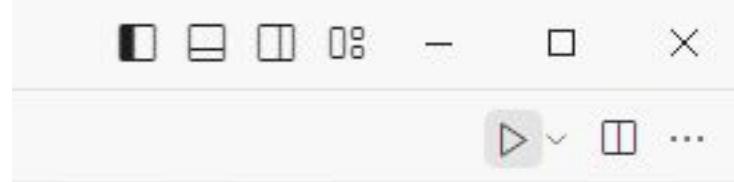


# Code Editor



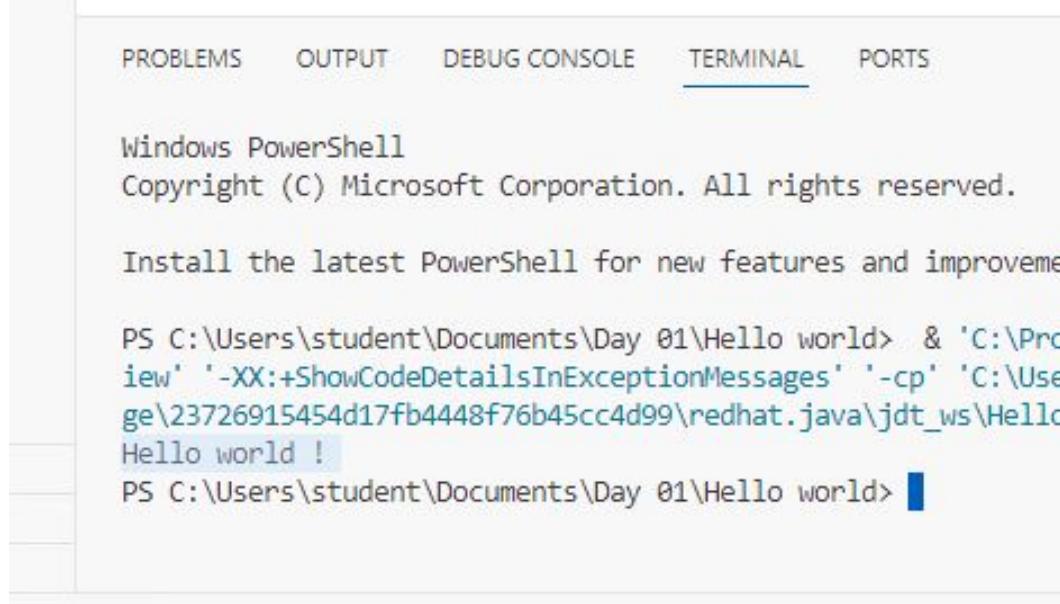
# Code Editor

Step 3: **Run** the program



# Code Editor

## Step 4: See the **output**



The screenshot shows the VS Code interface with the 'TERMINAL' tab selected. The terminal window displays the following output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improveme

PS C:\Users\student\Documents\Day 01\Hello world> & 'C:\Pro
iew' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Use
ge\23726915454d17fb4448f76b45cc4d99\redhat.java\jdt_ws\Hello
Hello world !
PS C:\Users\student\Documents\Day 01\Hello world>
```

# Variables

A variable is a data container.

It can contain 1 information per time.

```
int age = 30;
```

# Variables

```
data_type name = value ;
```

# Variables

- Start with letter or \_underscore
- Unique word in the code
- No special symbols
- No Java keyword

# Variables

```
public class VariableNaming {  
    public static void main(String[] args) {  
        // Correct variable names following the naming rules:  
  
        // Rule: Starts with a letter.  
        // This variable name starts with the letter 'a'.  
        int age = 30;  
  
        // Rule: Starts with an underscore.(but not recommended)  
        // This variable name starts with an underscore.  
        double _salary = 50000.0;
```

# Variables

## Use variable value

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

# Data Type

```
class DataTypes {
    public static void main(String [] args) {
        // integer types
        byte aSingleByte = 100; // -128 to 127
        short aSmallNumber = 20000; // -32,768 to 32,767
        int anInteger = 2147483647; // -2147483648 to 2147483647
        long aLargeNumber = 9223372036854775807L; // -92233720368547

        // decimal types
        double aDouble = 1.79769313; // 4.9E-324 to 1.797693
        float aFloat = 3.4028F; // 1.4E-45 to 3.40282

        // booleans
        boolean isWeekend = false;
        boolean isWorkday = true;

        // characters
        char copyrightSymbol = '\u00A9';

        System.out.println("This is the copyright symbol: " + copyrightSymbol);
        // This is the copyright symbol: ©

    }
}
```

# Let's start with simple data types

```
public class HelloWorld {  
    public static void main(String[] args) {  
        String name = "Lisa";  
        char lastNameFirstLetter = 'H';  
  
        int year_of_birth = 1996;  
        double height = 1.57;  
  
        boolean hasPets = true;  
  
        System.out.println("Hello, my name is " + name);  
    }  
}
```

# Operations

You can also do simple mathematical operations, like this:

```
int number1 = 10;  
int number2 = 15;  
int sum = number1 + number2;  
System.out.println(sum);
```

# Operations

```
1 public class BasicOperations {  
2     public static void main(String[] args) {  
3         // Arithmetic Operations  
4         int a = 10;  
5         int b = 5;  
6  
7         int addition = a + b; // Addition  
8         int subtraction = a - b; // Subtraction  
9         int multiplication = a * b; // Multiplication  
10        int division = a / b; // Division  
11        int modulus = a % b; // Modulus (remainder of division)  
12  
13        System.out.println("Addition: " + addition);  
14        System.out.println("Subtraction: " + subtraction);  
15        System.out.println("Multiplication: " + multiplication);  
16        System.out.println("Division: " + division);  
17        System.out.println("Modulus: " + modulus);  
18    }  
19 }
```

# Operations

```
1 public class BasicOperations {  
2     public static void main(String[] args) {  
3         // Arithmetic Operations  
4         int a = 10;  
5         int b = 5;  
6  
7         // Increment and Decrement  
8         int increment = ++a; // Increments and then uses the variable  
9         int decrement = --b; // Decrements and then uses the variable  
10  
11         System.out.println("Increment: " + increment);  
12         System.out.println("Decrement: " + decrement);  
13     }  
14 }
```

# Operations

```
1 public class BasicOperations {  
2     public static void main(String[] args) {  
3         // Arithmetic Operations  
4         int a = 10;  
5         int b = 5;  
6  
7         // Relational Operations  
8         boolean isGreater = a > b; // Greater than  
9         boolean isLess = a < b; // Less than  
10        boolean isEqual = a == b; // Equal to  
11  
12        System.out.println("Is a greater than b? " + isGreater);  
13        System.out.println("Is a less than b? " + isLess);  
14        System.out.println("Is a equal to b? " + isEqual);  
15    }  
16 }
```

# Operations

```
1 public class BasicOperations {  
2     public static void main(String[] args) {  
3         // Arithmetic Operations  
4         int a = 10;  
5         int b = 5;  
6  
7         // Logical Operations  
8         boolean andOperation = (a > 5) && (b < 10); // Logical AND  
9         boolean orOperation = (a > 10) || (b < 10); // Logical OR  
10        boolean notOperation = !(a == b); // Logical NOT  
11  
12        System.out.println("Logical AND result: " + andOperation);  
13        System.out.println("Logical OR result: " + orOperation);  
14        System.out.println("Logical NOT result: " + notOperation);  
15    }  
16 }
```

# Operations

## Math library

- Predefined constants (PI)
- Functions (exponential, **power**, etc)
- Utility methods (round, random, etc)

```
public static void main(String[] args) {  
    double P = 1000; // Principal amount  
    double r = 0.05; // Annual interest rate (5%)  
    int n = 12; // Compounded monthly  
    int t = 5; // Time in years  
  
    double A = P * Math.pow(1 + (r / n), n * t);
```

# **Now YOUR TURN !**

Let's do exercises number 2

# Input

Input is the process of providing data or information to a computer program during its execution.

# Input

To obtain an input in java, you need a special “tool” called **Scanner** that can recognize the entered data.

# Input

```
1 import java.util.Scanner;  
2  
3 public class BasicOperations {  
4     public static void main(String[] args) {  
5         Scanner scanner = new Scanner(System.in);  
6     }  
7 }
```

# Input

```
1 import java.util.Scanner;  
2  
3 public class BasicOperations {  
4     public static void main(String[] args) {  
5         Scanner scanner = new Scanner(System.in);  
6  
7         // getting data according to their type  
8         String name = scanner.nextLine(); // String  
9         int age = scanner.nextInt(); // int  
10    }  
11 }
```

# **Now YOUR TURN !**

Let's do exercises number 3

# Project Students - Step 1

Your program should store the following pieces of information for a student:

- First Name - String
- Last Name - String
- Birthday (Day of the Month) - int
- Birth Month - int
- Birth Year - int
- Course Registered - String

# Project Students - Step 1

The program should print the requested information plus the student's age.

For now, we don't take month and day into account for the calculation.

To get the current year:

```
LocalDate today = LocalDate.now();
int currentYear = today.getYear();
```

Output:

```
Enter first name: Ana
Enter last name: Gaggero
Enter birthday (day of month): 22
Enter birth month: 10
Enter birth year: 1982
Enter course registered: Java
```

```
Student Name: Ana Gaggero
Date of Birth: 22/10/1982
Age: 42
Course Registered: Java
```

# Project Tic Tac Toe - Step 1

- Ask the names of both players
- Print the board before the first move

```
What is your name Player 1? Ana  
What is your name Player 2? Juan
```

```
Ana will be X and Juan will be O
```

1	2	3
4	5	6
7	8	9