HelloWorld.java

Program Breakdown
(Printing to Console)
Problem Statement

Write a program called HelloWorld.java that will execute code to print the text "Hello, World!" to the console output.
Problem Statement

Write a program called HelloWorld.java that will execute code to print the text "Hello, World!" to the console output.

What we need:

- Way to specify/write our program
- Way to execute code
- Way to print text to the console
- A way to represent the text "Hello, World!"
public class HelloWorld {

}

**Keywords:**
**You must spell these words exactly and write them in the correct order so Java recognizes them.**
- These words (public class) indicate (declare) to Java that you are writing a class that can be accessed for, among other things, running code.
- A class is a container for the code of a program
- There can be only one public class per file

**Specifying the name of the program:**
- You can name a class whatever you want as long as:
  - It contains only letters, numbers, underscores, and dollar sign symbols
  - Must begin with a letter
- The name of the class must exactly match the name of the file, so this class must be saved in a file called HelloWorld.java.

**Scope:**
- Everything between the curly braces is part of the program; it is within the “scope” of the class HelloWorld
- This is where the code for the program goes
Problem Statement

Write a program called HelloWorld.java that will execute code to print the text "Hello, World!" to the console output.

What we need:

- Way to specify/write our program
- Way to execute code
- Way to print text to the console
- A way to represent the text "Hello, World!"

- class
- ???
Main method declaration

```java
public class HelloWorld {
    public static void main(String[] args) {
        // Everything between the curly braces is part of (within the scope of) the main method.
        // This is where the executable statements of code go.
        // Java will execute the statements in main in order from top to bottom.
    }
}
```

**Keywords:**
- These words (public static void) indicate (declare) to Java that you are writing a method that contains statements of executable code.
- A method is a subroutine, a part of the overall procedure of the program that has been labeled.

**Specifying main method:**
- Using the declaration “main” with “String[] args” between the parentheses will indicate to Java that this is the method that should be executed when the program is run.
- For now, remember to type “main(String[] args)” exactly for declaring the method.
Problem Statement

Write a program called HelloWorld.java that will execute code to print the text "Hello, World!" to the console output.

What we need:

- Way to specify/write our program
- Way to execute code
- Way to print text to the console
- A way to represent the text "Hello, World!"

- class
- main
- ???
System.out.println() statement

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

**System.out.println:**
- Executable call – asks Java to print whatever is between the parentheses, then move print cursor to next line
- If the parentheses are empty like `System.out.println()`, we will just print a blank line
- If we want to print text, we need to specify what text we want printed on the line in between the parentheses

**Statement:**
- The semicolon (;) indicates to Java that whatever comes before it is a statement of code to be executed
Problem Statement

Write a program called HelloWorld.java that will execute code to print the text "Hello, World!" to the console output.

What we need:

- Way to specify/write our program
- Way to execute code
- Way to print text to the console
- A way to represent the text "Hello, World!"

- class
- main
- System.out.println();
- ???
String literal

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

String literal:
- To represent text in Java, we use two quotation marks and put the characters in the text between the quotation marks – we call this a String (of characters) and we say it is a String literal when we are dealing with the quotes
- If we want to print Hello, World!
  We ask for System.out.println with the String literal “Hello, World!”
Problem Statement

Write a program called HelloWorld.java that will execute code to print the text "Hello, World!" to the console output.

What we need:

- Way to specify/write our program
- Way to execute code
- Way to print text to the console
- A way to represent the text "Hello, World!"

- class
- main
- System.out.println();
- String literal