TASK 1: Java Environment Setup & First Executable Program

STEP 1: What is Java?

Java is a programming language.

Works:-

java file (source code)

↓ javac (compiler)

.class file (bytecode)

↓ JVM

Output on screen

Java does not run directly on the OS.
It runs on JVM, which makes Java platform independent.

STEP 2: Install JDK

What is JDK?

JDK (Java Development Kit) is required to:

- Write Java programs
- Compile Java programs
- Run Java programs

Without JDK, Java will not work.

How to Install JDK

1. Open browser
2. Go to https://adoptium.net
3. Download JDK 8 or higher
4. Install it normally (Next → Next → Finish)

STEP 3: Set JAVA_HOME (Why & How)

Why JAVA_HOME is Needed?

- So your system knows where Java is installed

- Allows you to use java and javac commands from anywhere

How to Set JAVA_HOME (Windows)

1. Search → Environment Variables
2. Click Edit system environment variables
3. Click Environment Variables

Add New System Variable:

Variable Name: JAVA_HOME

Variable Value: C:\Program Files\Java\jdk-17

Edit Path Variable:

Add: %JAVA_HOME%\bin

STEP 4: Verify Java Installation:

After install (JDK) , we can verify install version are correct & incorrect.

Open Command Prompt and type:

java -version

Correct Output Means:

- Java runtime works
- Java compiler works

Error Means:

- JAVA_HOME or Path is wrong

STEP 5: Create Java Project Manually

- Why Manual Folder Creation?

So you understand how Java projects are structured without IDE help.

Create folders like this:

```
Java_Task_1
 ├── src
 │    └── HelloJava.java
 ├── screenshots
 └── README.md
```

src → Source code

screenshots → Proof of execution

README.md → Explanation of work (for GitHub)

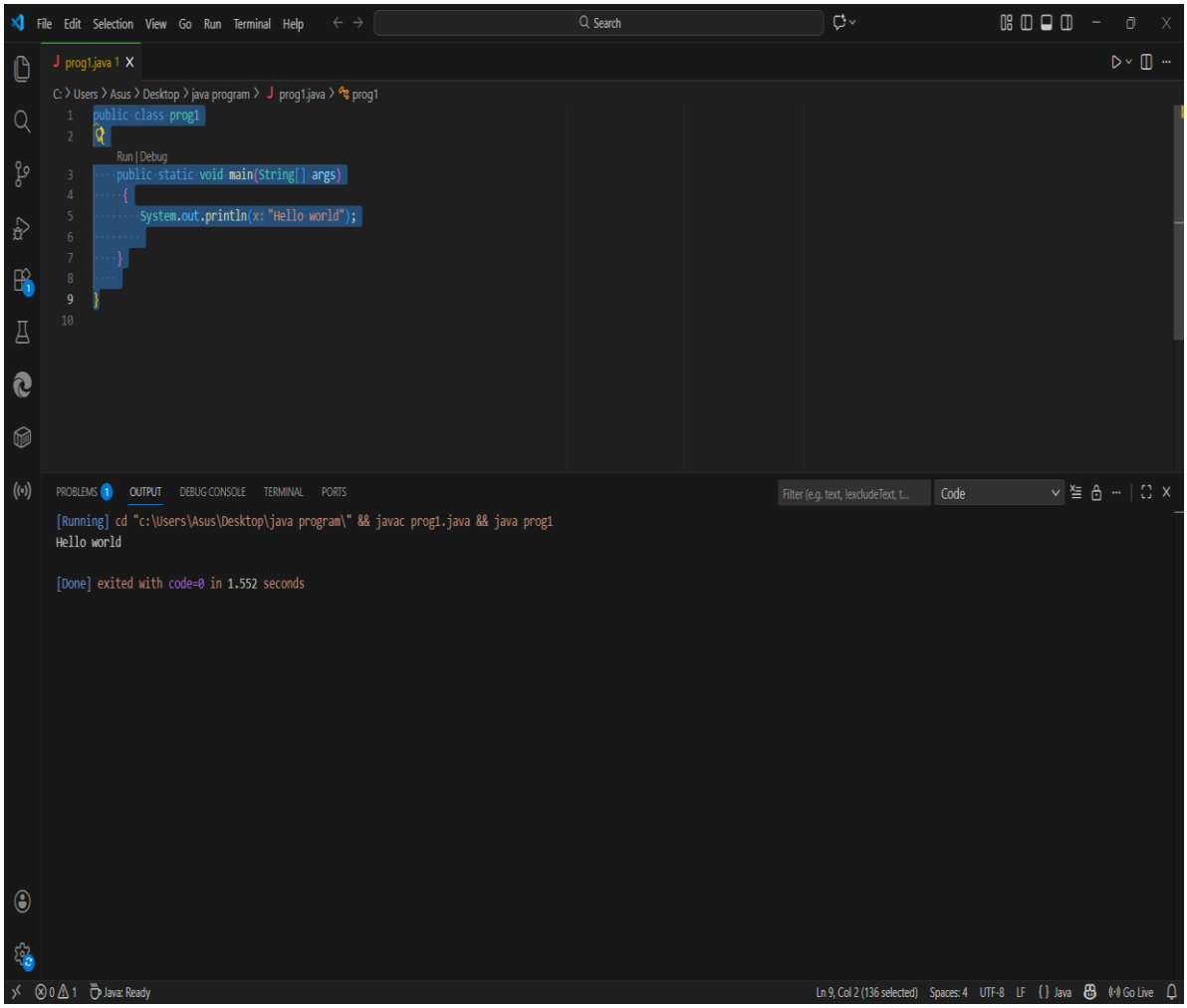
STEP 6: Write Java Program

Code:-

```
public class prog1
{
   public static void main (String[] args)
    {
      System.out.println("Hello world");

    }

}
```



Explanation:

class Prog1.java

- Every Java program must have a class

- Class name must match file name

  public static void main(String[] args)
- Entry point of Java program
- JVM starts execution from here

  Why static?
- JVM does not create object
- Static allows direct call

  String[] args
- Used to accept command-line arguments