22610026

Elizabeth Suresh Pawar

Assignment – 1

1. Describe JDK, JRE, JVM.
2. **JDK (Java Development Kit)**: JDK is a software development kit used by developers to create Java applications. It contains tools such as compilers, debuggers, and other utilities necessary for developing Java programs. The JDK also includes a set of class libraries and frameworks that provide core functionalities for Java development.
3. **JRE (Java Runtime Environment)**: JRE is an environment in which Java bytecode can be executed. It includes the Java Virtual Machine (JVM) and the core libraries required to run Java applications. JRE is primarily used by end-users who want to run Java applications on their systems but are not involved in the development process.
4. **JVM (Java Virtual Machine)**: JVM is an abstract computing machine that provides an execution environment for Java bytecode. It interprets the bytecode and executes the instructions on the native hardware platform. JVM acts as an intermediary between the compiled Java code and the underlying operating system, providing platform independence for Java programs. It also manages memory, performs garbage collection, and provides other runtime services necessary for executing Java applications.

2. Differentiate between C, C++, JAVA.

| **Feature** | **C** | **C++** | **Java** |
| --- | --- | --- | --- |
| Paradigm | Procedural | Object-Oriented | Object-Oriented |
|  |  |  | (with some functional aspects) |
|  |  |  | and concurrent programming |
| Platform | Cross-platform | Cross-platform | Cross-platform |
| Independence |  |  | (via JVM) |
| Memory Management | Manual | Mostly manual, with some | Automatic (Garbage Collection) |
|  |  | support for automatic |  |
|  |  | memory management (e.g., |  |
|  |  | smart pointers) |  |
| Usage | System programming, | System programming, | Web development, Enterprise |
|  | Embedded systems, | Game development, | software, Mobile apps (Android) |
|  | Performance-critical | High-performance |  |
|  | applications | applications |  |
|  |  |  |  |
| Features | - Low-level access to | - Object-oriented | - Object-oriented |
|  | system resources | programming | programming |
|  | - Pointers and manual | - Inheritance, | - Garbage collection |
|  | memory management | polymorphism, | - Exception handling |
|  | - Limited standard | encapsulation | - Multithreading |
|  | library | - Templates (generic | - Platform independence (JVM) |
|  |  | programming) |  |
|  |  | - Standard library |  |
|  |  |  | - Reflection |
|  |  |  | - Annotations |

3. Explain simple hello word program in Java. Explain compilation and

execution of java program.

// HelloWorld.java

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

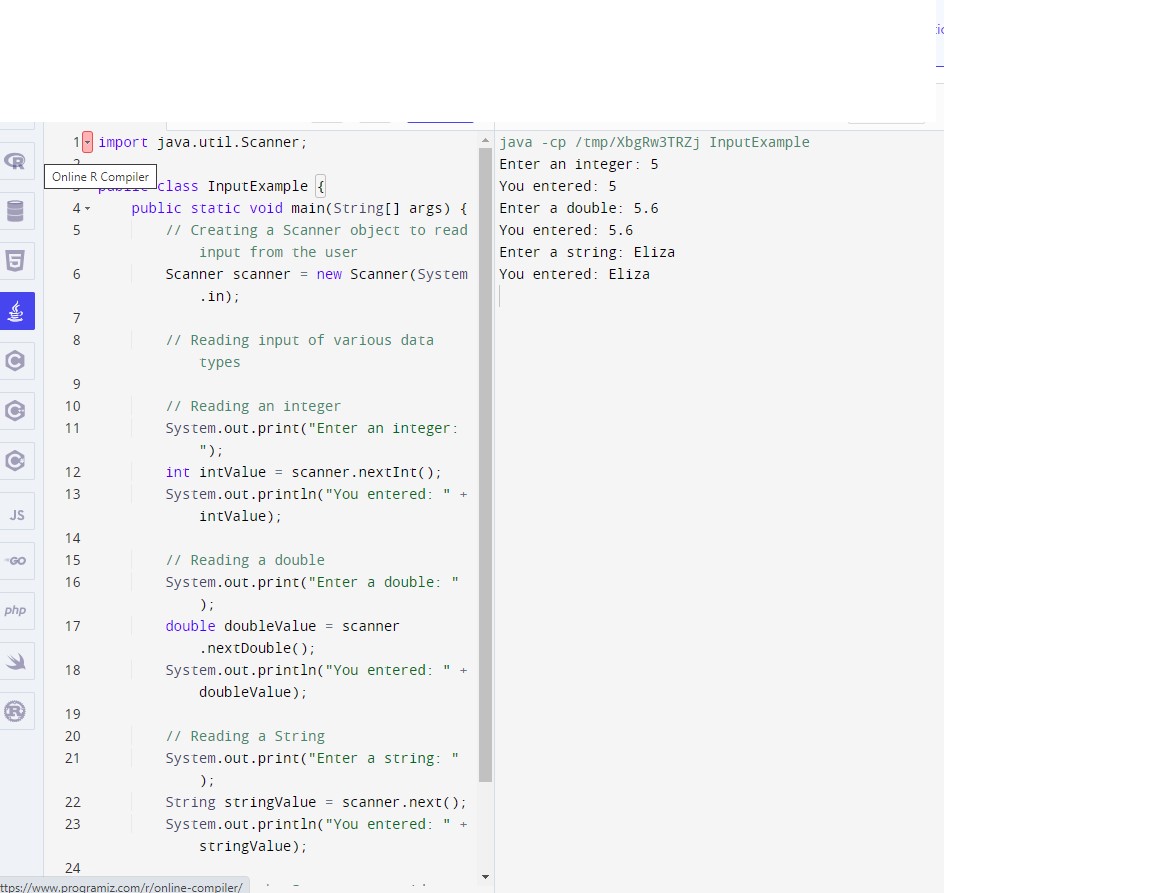
**Compilation**: Once you've written the Java source code, you need to compile it into bytecode. To do this, you use the Java compiler (**javac**), which is included with the Java Development Kit (JDK).

**Execution**: After successfully compiling the Java source code, you can execute the program using the Java Virtual Machine (JVM). To do this, run the following command:

java HelloWorld

4. Write java program for reading input of various data types from user

using scanner class.



1. Write a Java program to convert seconds to hour, minute and seconds.

public class SecondsConverter {

public static void main(String[] args) {

// Creating a Scanner object to read input from the user

Scanner scanner = new Scanner(System.in);

// Reading input in seconds

System.out.print("Enter the number of seconds: ");

int totalSeconds = scanner.nextInt();

// Calculating hours, minutes, and remaining seconds

int hours = totalSeconds / 3600;

int minutes = (totalSeconds % 3600) / 60;

 int seconds = totalSeconds % 60;

// Displaying the result

System.out

.println("Converted time: " + hours + " hours, " + minutes + " minutes, and " + seconds + " seconds.");

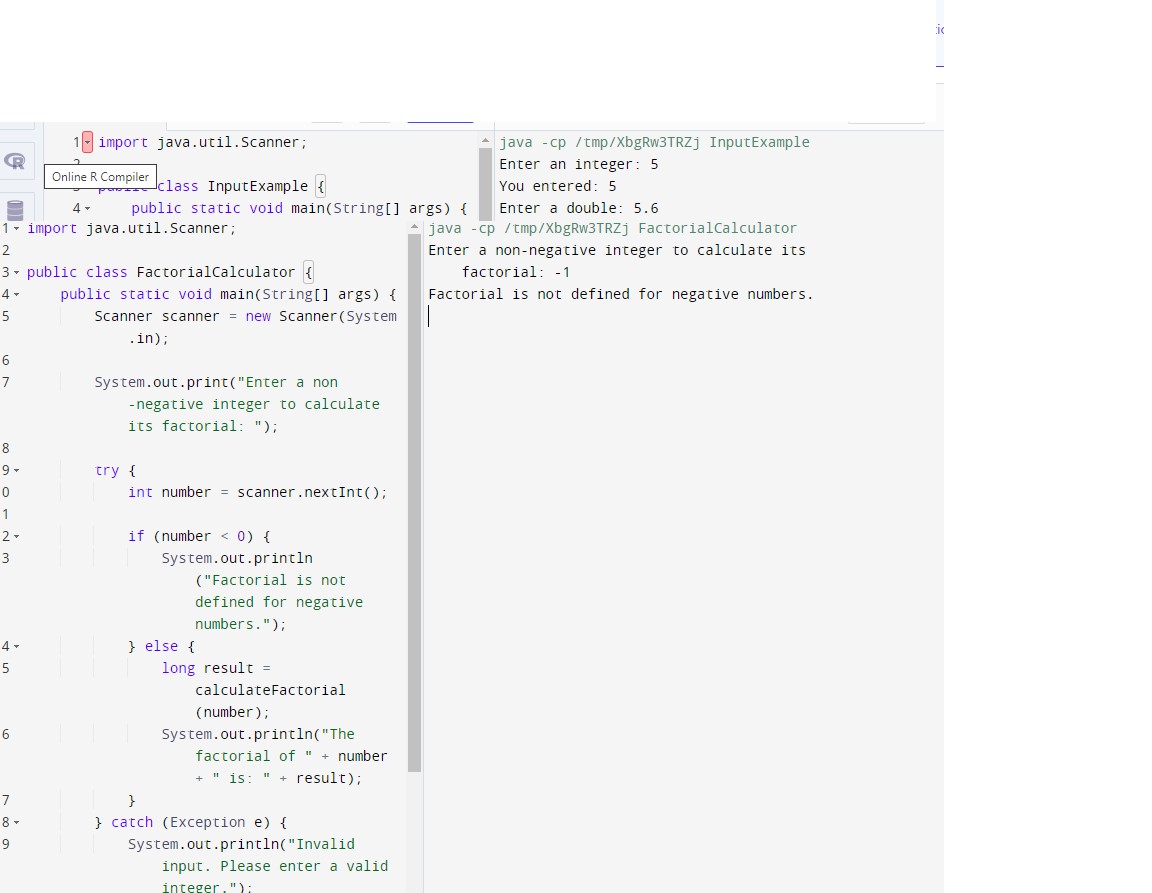
// Closing the Scanner to avoid resource leak

scanner.close();

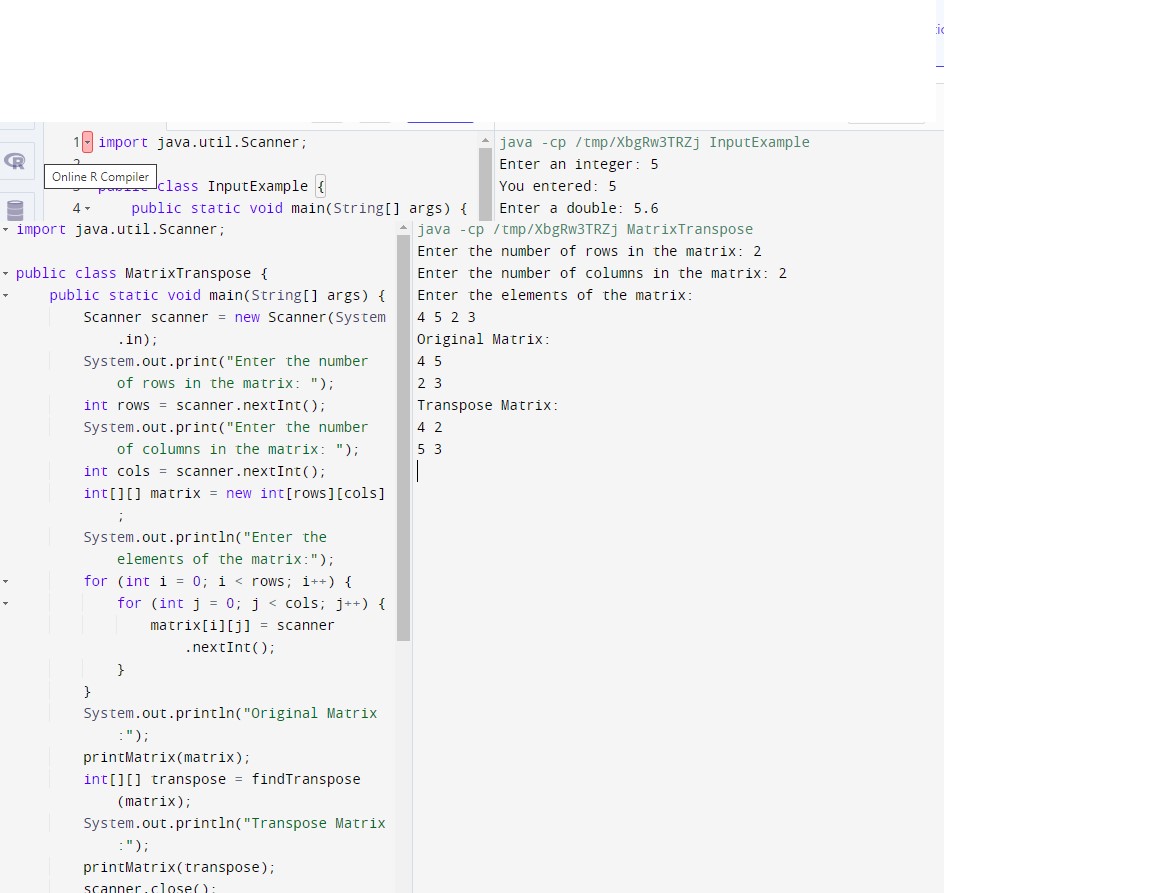
}

}

1. Write a Java program to check if there is a 10 in a given array of integer
2. Write a program to calculate the factorial of a number. (The number is passed as the command-line argument whose factorial we need to calculate)



1. Write a Java Program to find transpose of Matrix



1. Write a program to implement different types of constructors.
2. 