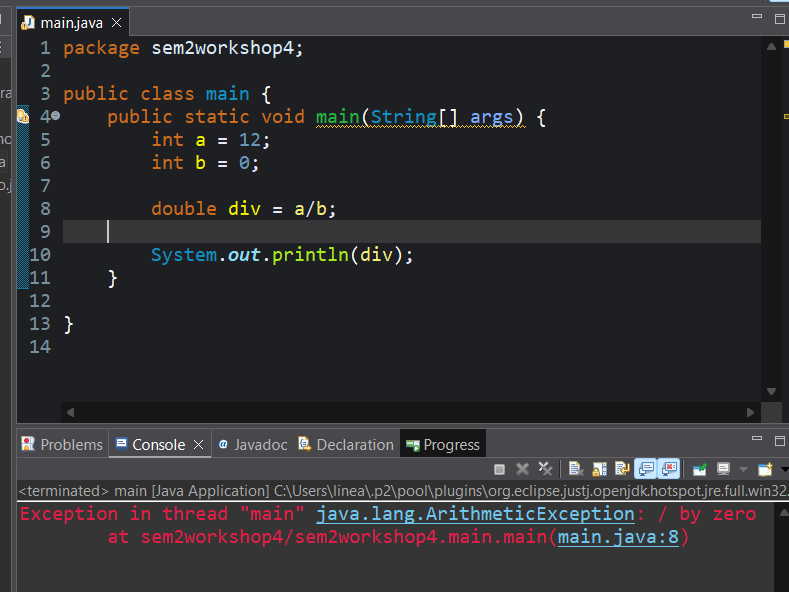
**Workshop 4**

**Exception Handling**

1. Write a Java program to demonstrate the difference between an Exception and an Error.

 A screen shot of a computer program

AI-generated content may be incorrect.

1. Create a program that demonstrates ArithmeticException by dividing a number by zero and handle it using try-catch.

A screen shot of a computer program

AI-generated content may be incorrect.

1. Write a Java program to show the complete Exception Hierarchy using comments and examples.
2. Develop a program to handle multiple types of exceptions (ArithmeticException, ArrayIndexOutOfBoundsException, and NullPointerException) in a single try block.

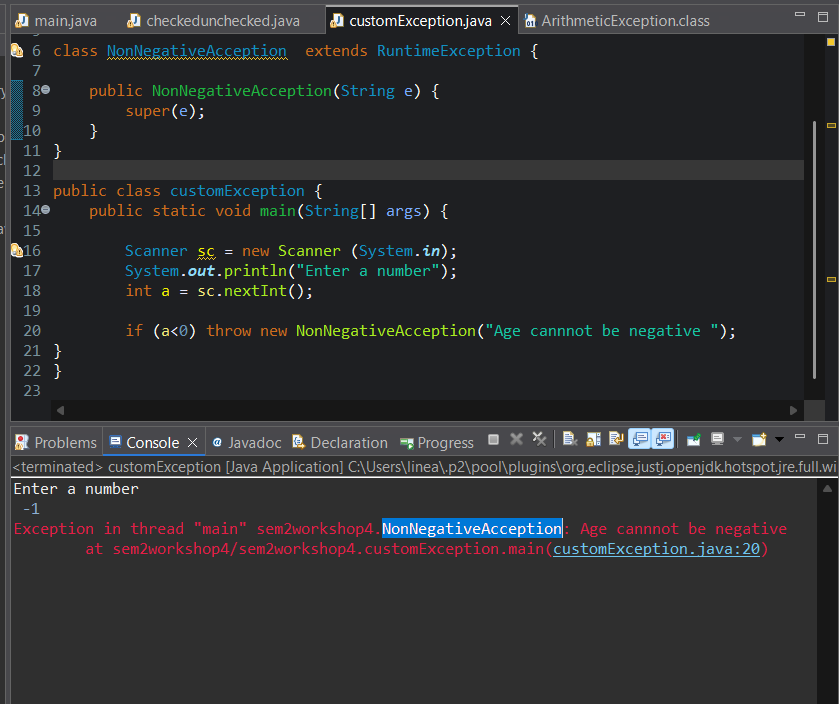
A screen shot of a computer program

AI-generated content may be incorrect.

1. Write a program that demonstrates checked and unchecked exceptions with suitable examples.



1. Create a program that reads an integer from the user and throws a custom exception if the number is negative.



1. Write a Java program using multiple catch blocks to handle different exceptions separately.

A screen shot of a computer program

AI-generated content may be incorrect.

1. Create a program demonstrating nested try-catch blocks.

A screen shot of a computer

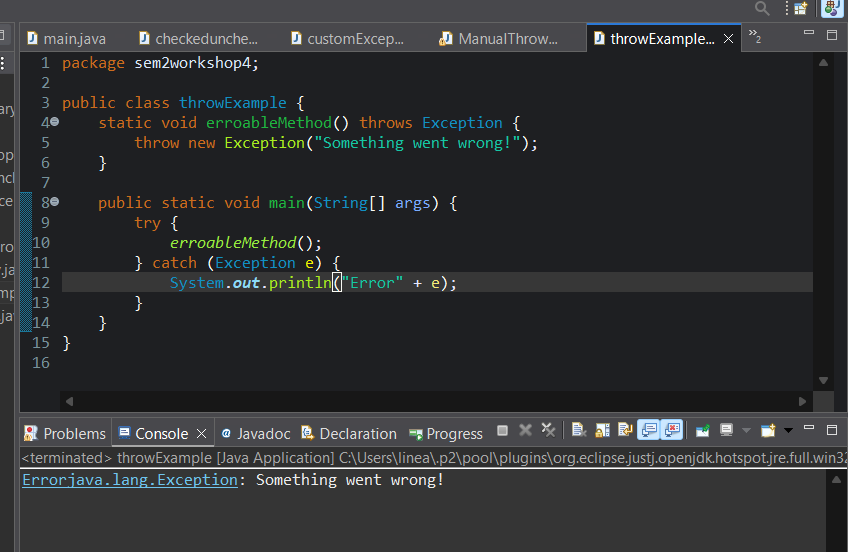
AI-generated content may be incorrect.

1. Write a Java program that uses the throw keyword to manually throw an exception when invalid input is entered.

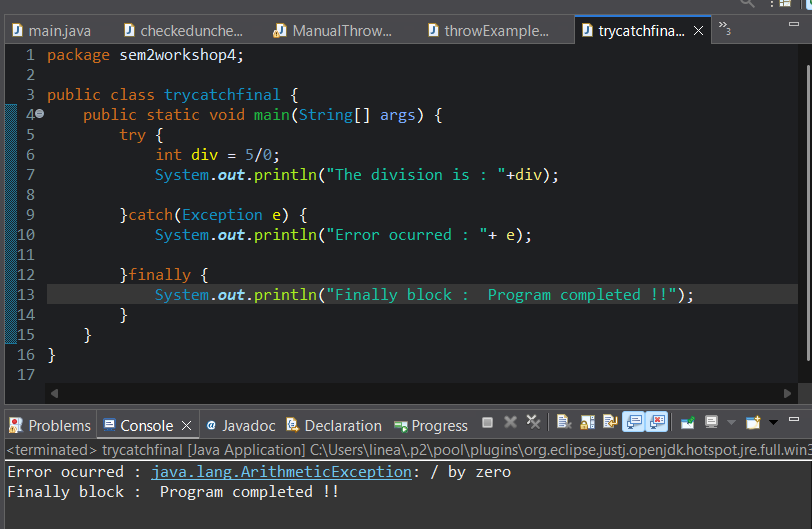
A screen shot of a computer program

AI-generated content may be incorrect.

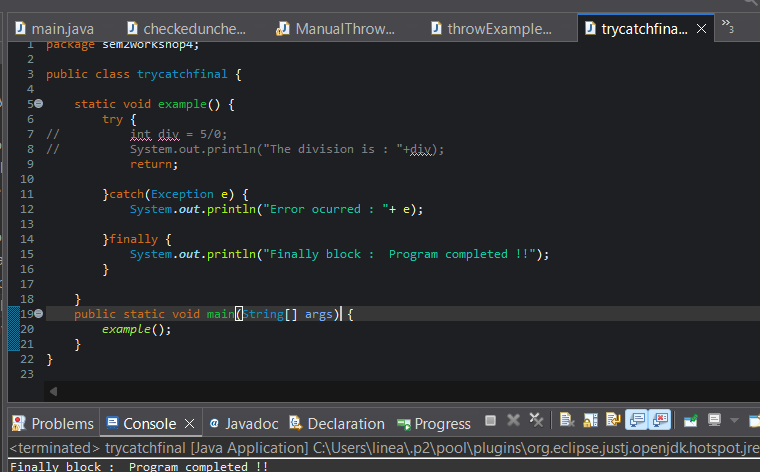
1. Develop a Java program that uses the throws keyword in a method and handle the exception in the calling method.



1. Write a Java program to demonstrate the use of the finally block in exception handling.



1. Create a program to show that the finally block executes even when return is used inside a try or catch block.



1. Write a program to demonstrate try-with-resources by reading data from a file safely.

A screen shot of a computer program

AI-generated content may be incorrect.

1. Develop a Java program to handle multiple exceptions using multi-catch syntax (catch(Exception1 | Exception2 e)).

A computer screen shot of a program

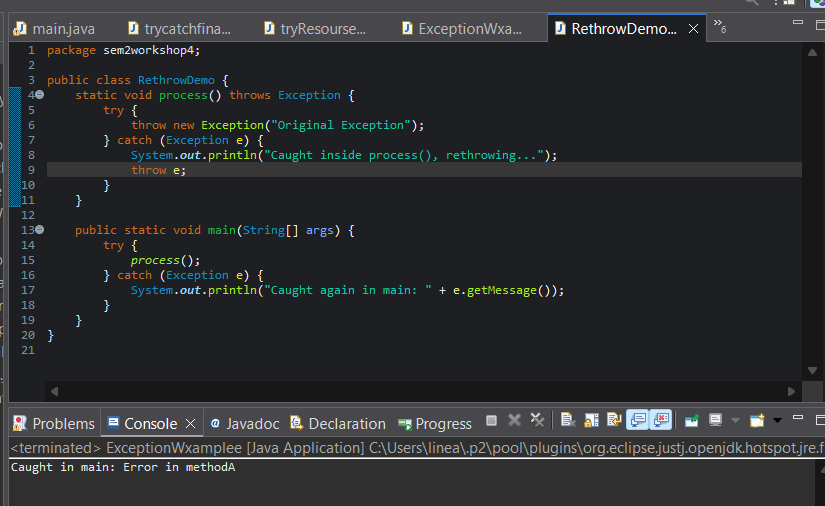
AI-generated content may be incorrect.

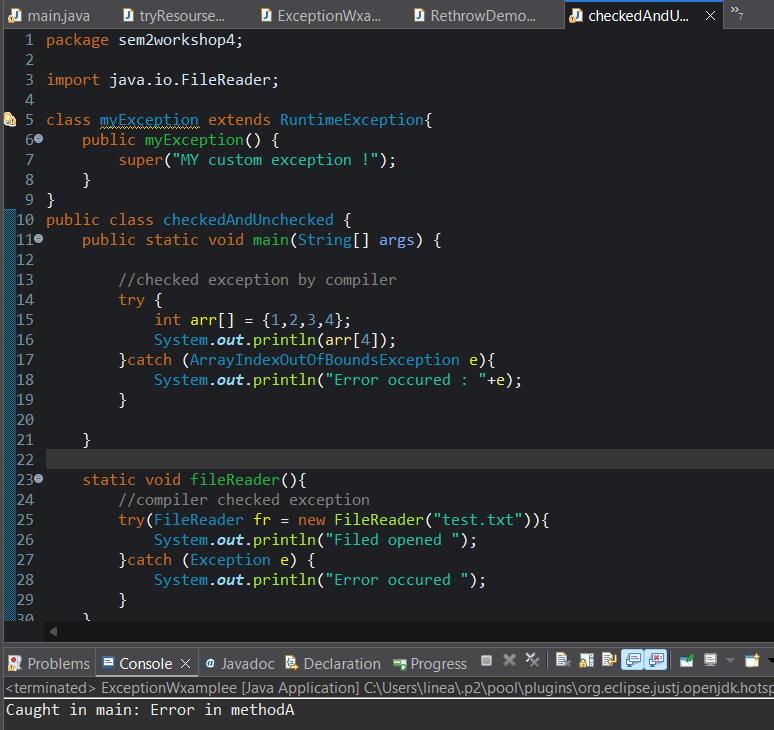
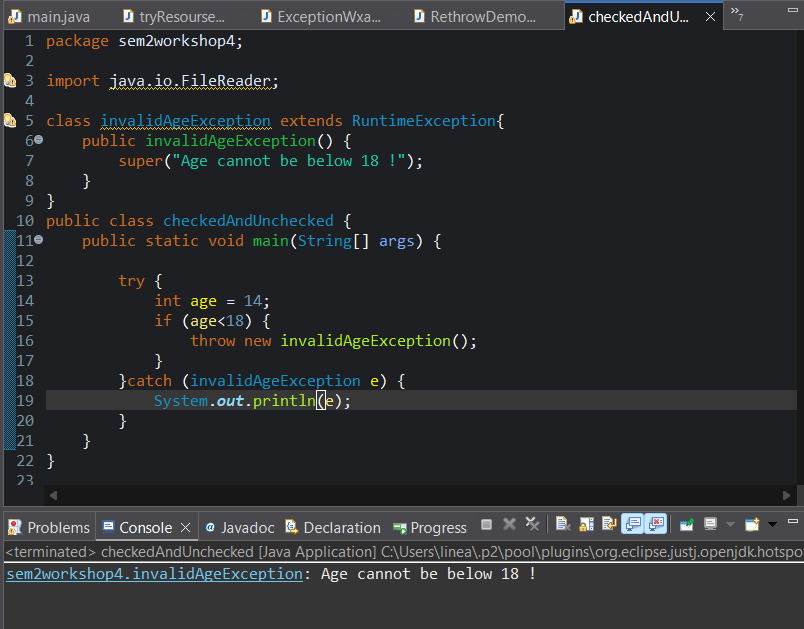
15.Write a program that shows how an exception is propagated from one method to another using throws.

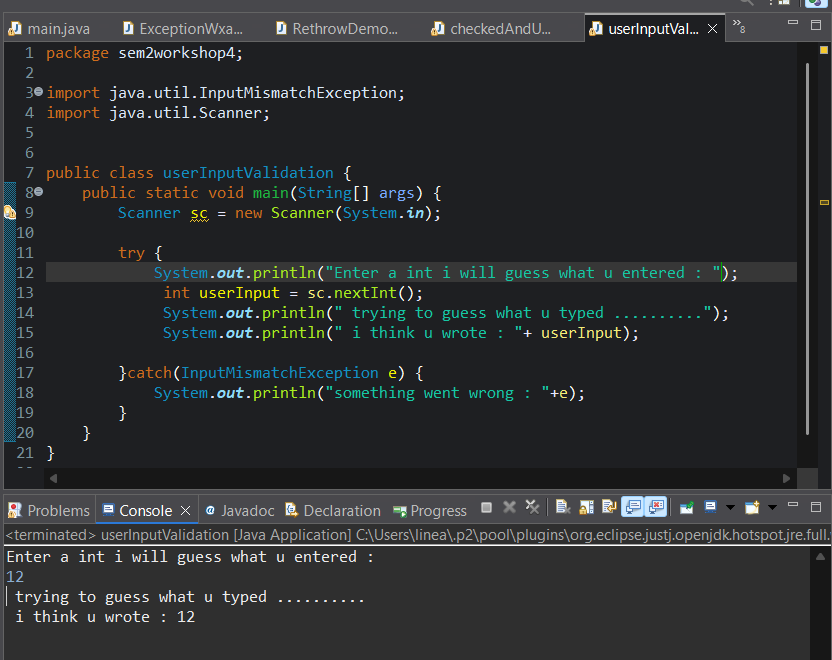
A screen shot of a computer program

AI-generated content may be incorrect.

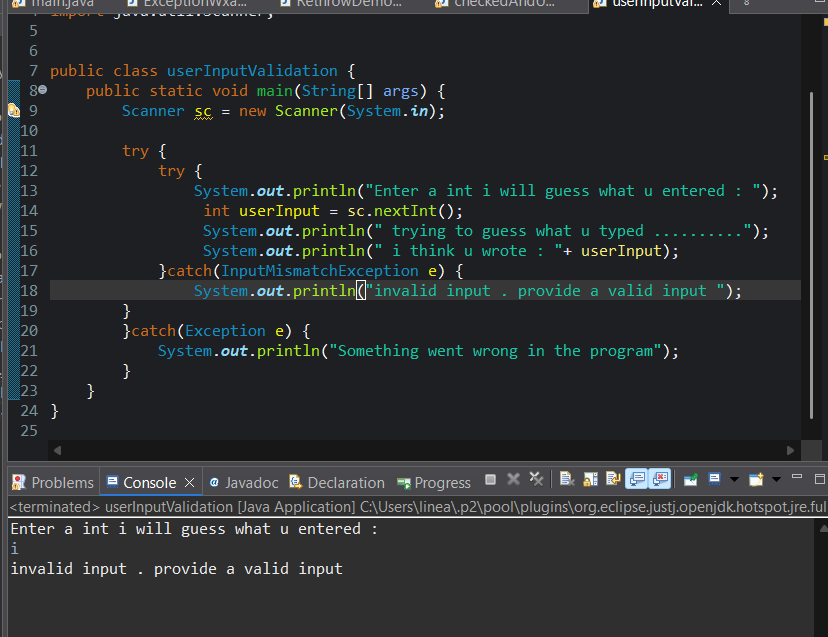
1. Create a program to demonstrate the concept of rethrowing an exception.



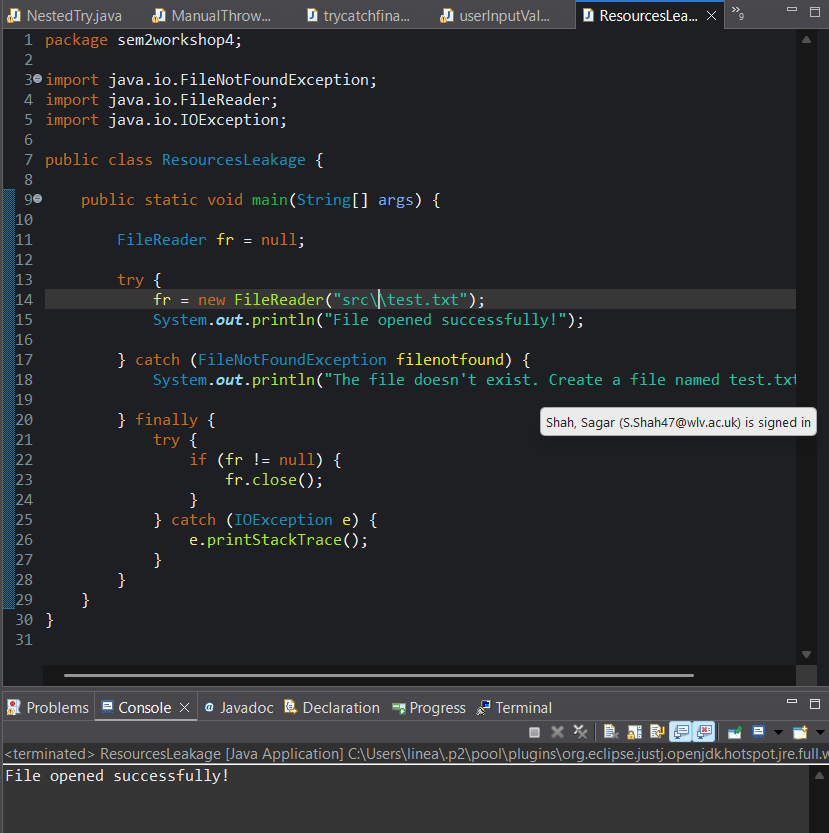
1. Write a Java program to demonstrate the difference between checked and unchecked exceptions by creating your own custom checked exception. 
2. Create a class InvalidAgeException and write a program that throws this exception if a person’s age is below 18. 
3. Write a Java program to handle user input errors gracefully using exception handling.

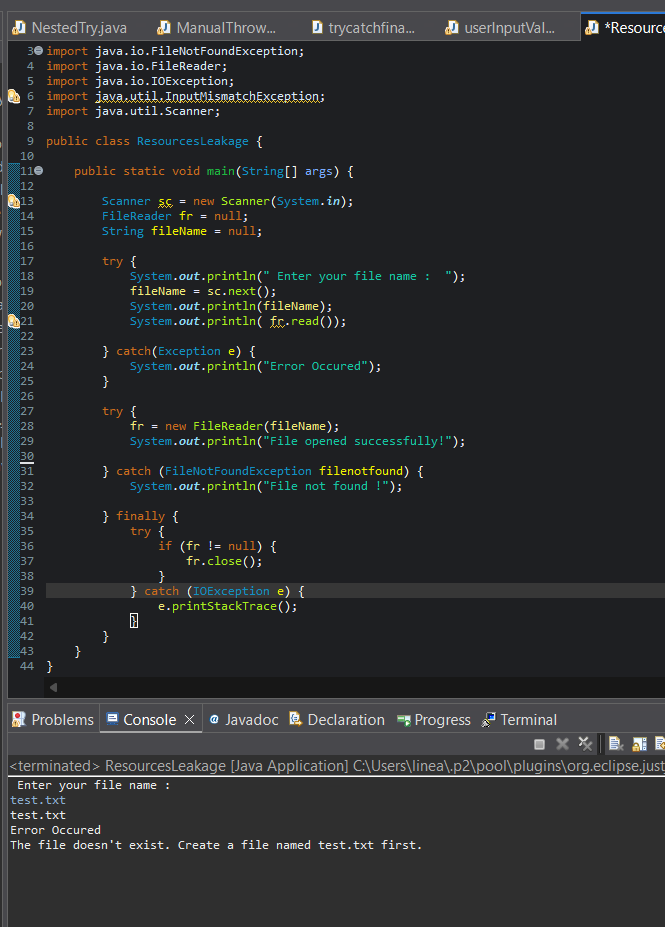


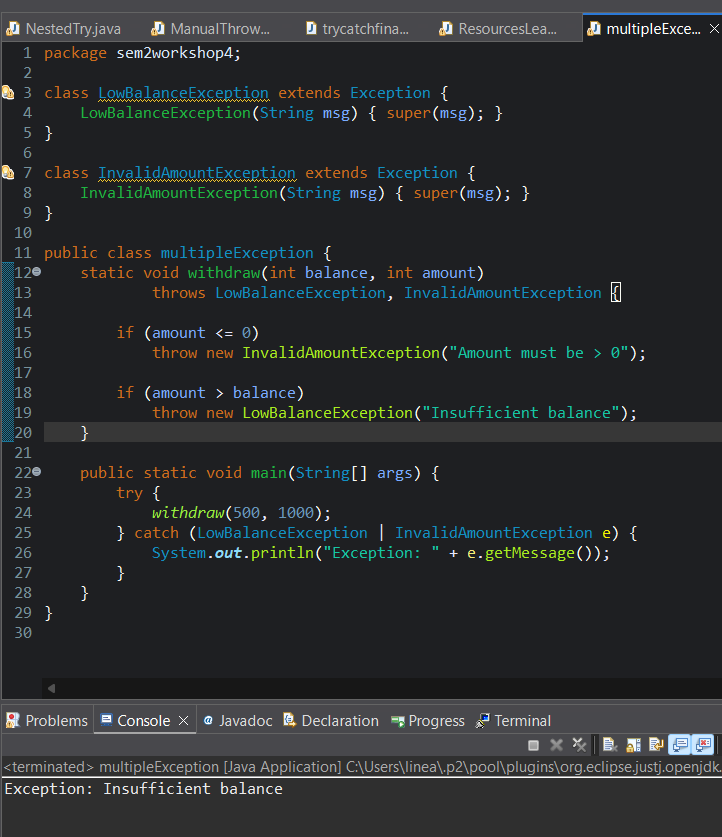
1. Develop a Java program to demonstrate exception chaining (new Exception("msg", cause)).



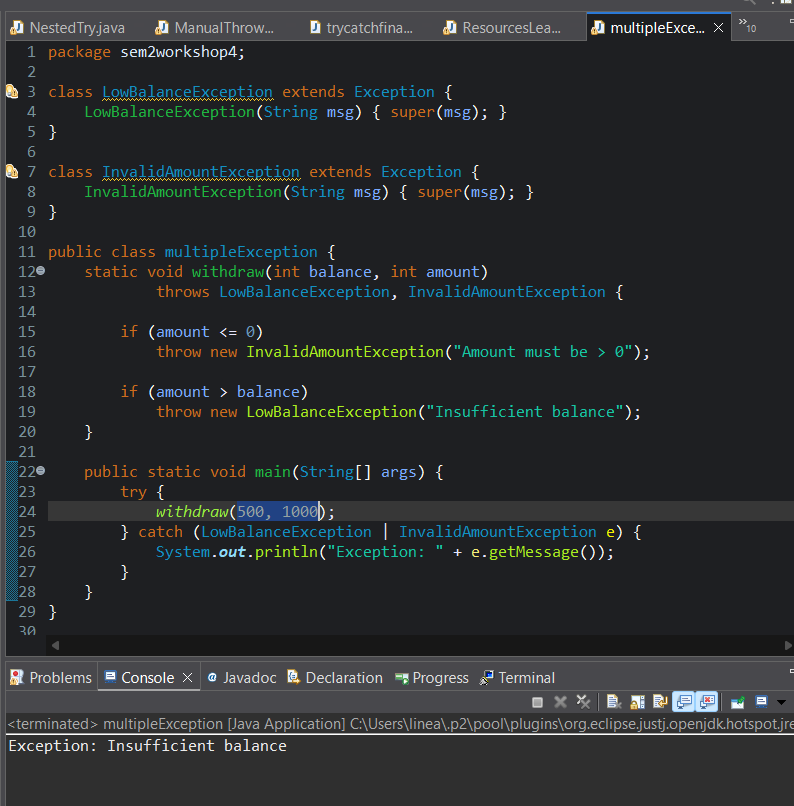
1. Create a program to demonstrate how finally is used for closing resources like files or database connections.



1. Write a program that takes a filename from the user, reads its content, and handles FileNotFoundException. 
2. Write a program that demonstrates how to create and handle multiple custom exceptions in a single program.



1. Create a Java program to demonstrate how exception handling improves program reliability and flow control.



1. Develop a Java program that demonstrates the difference between throw and throws with examples in different methods. 