Java Programming

2-4: Exceptions and Assertions

Practice Activities

Lesson Objectives:

- Use exception handling syntax to create reliable applications
- Recognize common exception classes and categories
- Create custom exception and auto-closeable resources
- Test invariants by using assertions
- Use try and throw statements
- Use the catch, multi-catch, and finally statements

Vocabulary:

Identify the vocabulary word for each definition below.

try-with-resources	A certain kind of try statement that contains resources.
Class invariants	An invariant used to evaluate the assumptions of the class instances.
Assertions	Certain types of boolean statements that allow you to test specific aspects of your code.
try	Key statement for handling exceptions in Java.
Internal invariants	An invariant that handles boolean statements to test internal values.
Control flow invariants	An invariant that handles conditions in control flow statements.
Multi-Catch Statement	A statement that allows you to handle multiple exceptions.
finally	An optional addition to a try-catch statement that will always be executed.
RuntimeException refers to	Run-time errors that can be handled inside the program.

handled exceptions (unchecked)

Try It/Solve It:

1. You have included exception handling for the create button in the JavaBank application. Do the same for the make transaction button.

   Done

2. Create an exception class called “myException” that accepts a String message as a parameter in its constructor and passes the message to the super class to be printed out when an error message is thrown.

3. Update all of the `catch(Exception e)`statements to create a MyException object named newExc that sends the message `"An unhandled error occurred!!"` to the console .

```
catch (Exception e) {
    e.printStackTrace();
```

4. Create a block of code that utilizes all three types of invariants and asserts their values.