Handling Exceptions

- An *exception* is an error affecting program execution
- If an exception is not *handled* the application will terminate abruptly
  - o Even if the application terminates, the exception handler can allow a graceful termination including an informative error message

Exception Handler: `try-catch-finally`

- A block of code that performs an action when an exception occurs
  - o `try` statements:
    - the statements that could possibly generate an exception
  - o `catch` clause:
    - waits for the exception matching the `exception` parameter and then executes its code
  - o `finally` clause:
    - optional
    - executes its statements regardless of what happens in the `try-catch` portion
- An exception handler is required when calling certain methods
  - o For ex) the createNewFile() method generates an IOException when the file name cannot be used to create a file (see the modified code below)

○

General form:

```
try {
    <statements>
} catch (exception err_code) {
    <statements>
} ... additional catch clauses
} finally (exception err_code) {
    <statements>
}
```

Modified TestFiles application (checks for existence of a file before creating a new one)

```java
import java.io.*;

/**
 * A program for demonstrating file objects.
 */
public class TestFiles {

    public static void main(String[] args) {
        File textFile = new File("c:\\supplies.txt");
        if (textFile.exists()) {
            System.out.println("File already exists.");
        } else {
            try {
                textFile.createNewFile();
                System.out.println("New file created.");
            } catch (IOException e) {
                System.out.println("File could not be created.");
                System.err.println("IOException: " + e.getMessage());
            }
        }
    }
}
```

Checks for the existence of a file before creating a new one.

Programming Exercise:

Alter your MyFile application so that it creates a file named zzz.txt and then displays a message indicating that the file has been creted. The application should prompt the user to either keep or delete the file. If the file is deleted, a message should notify the user when the file has been successfully deleted. Include a try-catch statement as in the example above.

Submit your source code to the Google Doc "ICS4U – Activity Submission Form"