Humna Sultan
Mr. Hendricks
AP CS A
December 11 2020

1/7/21 Notes Work

11.1: Reading and Writing Text Files

- To read input from a disk file, first construct a FileReader object with the name of the input file, then use the FileReader to construct a Scanner object:
- When reading text files, use the Scanner class.
- When writing text files, use the PrintWriter class and the print/println methods.
- You must close all files when you are done processing them.

11.2: Throwing Exceptions

- There are two main aspects to exception handling: reporting and recovery.
- In Java, exception handling provides a flexible mechanism for passing control from the point of error reporting to a competent recovery handler.
- To signal an exceptional condition, use the throw statement to throw an exception object.
- When you throw an exception, the current method terminates immediately.

11.3: Checked and Unchecked Exceptions

- There are two kinds of exceptions: checked and unchecked. Unchecked exceptions extend the class RuntimeException or Error.
- Checked exceptions are due to external circumstances that the programmer cannot prevent. The compiler checks that your program handles these exceptions.
- Add a throws specifier to a method that can throw a checked exception.

11.4: Catching Exceptions

- In a method that is ready to handle a particular exception type, place the statements that can cause the exception inside a try block, and the handler inside a catch clause.
- When the catch (IOException exception) block is executed, then some method in the try block has failed with an IOException.
- It is important to remember that you should place catch clauses only in methods in which you can competently handle the particular exception type.

11.5: The Finally Clause

- The finally construct is used to take some action whether or not an exception is thrown.
- It is important to close a PrintWriter to ensure that all output is written to the file.
- Once a try block is entered, the statements in a finally clause are guaranteed to be executed, whether or not an exception is thrown.

11.6: Designing your Own Exception Types

- Sometimes none of the standard exception types describe your particular error condition well enough, so you can make your own exception class.
- You can design your own exception types—subclasses of Exception or RuntimeException.
- It is customary to provide two constructors for an exception class: a default constructor and a constructor that accepts a message string describing the reason for the exception.

11.7 - Case Study: A Complete Example

- The finally clause ensures that the file is closed even when an exception occurs.
- The throws specifier of the readFile method need not include the FileNotFoundException class because it is a subclass of IOException.
- To see exception handling at work, you can look at specific error scenarios.