

Lab 1 -- Java Exceptions and I/O

1. Java Exceptions and I/O

1. Find the local documentation for the Java 1.5.0 SDK. Bookmark it. Refer to it often. (The primary source is at Sun <http://java.sun.com/j2se/1.5.0/docs/>)
2. Find the documentation for the Java I/O platform inside the SDK docs. How many top-level interfaces and classes does `java.io` contain?
3. Create a text file called `fred.txt` in your file space using an editor (Notepad will do). Enter a few lines of text and save the file. Now write a Java program that uses the `File` class to create a new file instance using the path name of your file. Check whether your application can read the file (what is the obvious method to use for this?).
4. Use the `File` class to: 1) create a directory in your file space; 2) print to the console a list of files and directories contained in a directory that exists in your file space; 3) prints a list similar to the previous one but including only the names of directories.
5. Open `fred.txt` as a `FileInputStream`. How can you do this 1) using a `File` object; 2) using a string containing the file name. Implement both approaches. Read the contents of `fred.txt` and print them to the console.
6. Repeat the previous exercise but open the file as a `FileReader`. What is the difference in these approaches?
7. Open a `DataOutputStream` to a new file called `numbers.bin`. Output the integers from 100 to 280 to the file. Close the file. Open the file as a `DataInputStream`. Read the file and display the integers on the console.
8. Use a `GZIP` filter stream to compress `numbers.bin` and write the output to a new file `numbers.bin.gz`. What percentage reduction in file size is achieved? Make sure that you can decompress and read `numbers.bin.gz`.

Note:

Write all your programs using exception handling and create test conditions to exercise the exception handling code.