Classes for Input/Output

(Save this for Reference!!)

To open the standard input stream for reading keyboard input use

```
EasyReader console = new EasyReader();
```

`console` is the name you give to the input stream (can be anything you like).

To open a text file for reading use

```
EasyReader inFile = new EasyReader(fileName);
```

`inFile` is the name you give to the input stream associated with the file (can be anything you like); `fileName` is a `String`.

Call the `bad()` method to check the status of the file. It returns `true` if the file is not opened properly or if there is an error or end of file; `false` otherwise. For example:

```
EasyReader inFile = new EasyReader(fileName);
if (inFile.bad())
{
  System.err.println("Can't open " + fileName);
  System.exit(1);
}
```

Examples for reading data from the keyboard or a file:

```
int n = console.readInt();        // reads an integer
double x = console.readDouble();  // reads a double

char ch = inFile.readChar();      // reads any character,
                                  //   including whitespace
String word = inFile.readWord();  // reads a contiguous string of
                                  //   non-whitespace characters

// Read and process all lines from a file:
String line;
while ((line = inFile.readLine()) != null)
{
    // process line:
    ...
}
```

Note:
`readInt`, `readDouble`, `readChar`, and `readWord` methods do not consume the end of the line after reading the last item. Call `readLine` to get rid of the tail of the line (even if only the `newline` character is left) before calling `readLine` on the next line.

Call `inFile.close()` to close the file.

To open a text file for writing use

```
EasyWriter outFile = new EasyWriter(fileName);
```

(or

```
EasyWriter outFile = new EasyWriter(fileName, "app");
```

if you want to append data to an existing file). `outFile` is the name you give to the output stream associated with the file (can be anything you like); `fileName` is a `String`.

Be careful:

`new EasyWriter(fileName)` wipes out the contents of the file if it already exists.

Call the `bad()` method that returns `true` if the file is not created (or opened for appending) properly; `false` otherwise.

Use `print` and `println` methods, the same as in `System.out`, to write data to a file. For example:

```
outFile.print("x = ");
outFile.println(x);
```

or

```
outFile.println("x = " + x);
```

Call `outFile.println()` to write a blank line.
Call `outFile.close()` to close the file.