File I/O PRACTICE

★ Work in groups of 2 — 3 ★

♣ You have to know how to do these things.
♠ I'll check your work, but won't be marking it.

1a. Read from keyboard	Use Scanner	
1b. Read from keyboard	Use Buffered Reader (wrapped around other IO objects)	
2a. Read from file	Use Scanner	Close scanner
2b. Read from file	Use Buffered Reader	Close the object afterwards
3. Write to file	Use PrintWriter	Close it!
4. Read from webpage (URL)	Use BufferedReader	Close it.
	Search for all lines with a certain word and print them out	
5. Read from keyboard using #1b and write the inputs to a file using #3		

- All of these steps should read/write multiple lines (ie. your file should have at least 3 lines in it, and you should be able to read the file until you get to the end).
- When you are using a file, then you'll need a file object.
  File f = new File(filename);
- Files and any other IO object need to be closed when you're finished with them.
- Please make your errors intelligible. If there's a FileNotFound exception, then state which file could not be found:
  System.out.println("Error: " + filename + " could not be found");
- When there's an IO-type exception, YOU have to decide what you want the program to do, to continue or not. e.g. if the file is not found, will the rest of the program work okay, or do you need "System.exit(0)" ?
- A summary is on next page, but you still have to get the try-catch stuff working, as well as closing resources, and also while loops to read multiple lines.

Useful summary from my notes (feel free to suggest changes)

read keyboard using Scanner

```
    Scanner keyb = new Scanner(System.in));
```

read keyboard using BufferedReader

```
    BufferedReader input = new BufferedReader (new InputStreamReader(System.in));
```

read file using Scanner

```
    Scanner input = new Scanner (new File(filename));
```

read file using buffered reader

```
    BufferedReader brFile = new BufferedReader (new FileReader (f));
```

read URL

```
    URL myURL = new URL("http://www.yahoo.com/");
    URLConnection urlConn = myURL.openConnection();

BufferedReader brWeb = new BufferedReader( new InputStreamReader( urlConn.getInputStream() ));
```

read URL (simpler)

```
    BufferedReader brWeb = new BufferedReader(new InputStreamReader(myURL.openStream()));
```

write to file (PrintWriter)

```
    PrintWriter pwFile = new PrintWriter( new BufferedWriter( new FileWriter (f)));
```

with flushing instead of buffering

```
    PrintWriter pwFile = new PrintWriter( new FileWriter (f), true);
```

write to file (BufferedWriter)

```
    BufferedWriter bwFile =  new BufferedWriter( new FileWriter (f)));
```

Location of files

```
System.out.println(  System.getProperty("user.dir") );
```

Or use absolute path names (but then it won’t work on other computers)

Methods

The final IO object that you create, the one that wraps all of the others, determines which methods you use to read/write the data.

```
Scanner:
          nextInt(), next(), nextLine(), ...

BufferedReader extends Reader:
          read()  -- reads a character (as int).
          readLine()

PrintWriter extends Writer:
          print(), println(), printf()
          write()

BufferedWriter extends Writer:
          write()

FileWriter extends Writer:
          write()
```