CSCI 161 Exam 2 Study Guide

This study guide is provided for your review. Please come prepared to the exam knowing the material and able to do the types of problems reviewed in this guide.

1) Know the following glossary of terms and what each means:

boilerplate code
boolean
char
Character class
cumulative algorithm
current directory
fencepost algorithm
file
File class
file path
Integer class
kludge
line-based processing
printf
relative path
roundoff error
sentinel loop
short-circuit evaluation
text processing
token-based processing

2) Know the pseudocode for Cumulative Sum, Min, Max, and Sentinel loops

3) Be able to describe the traits of a fencepost algorithm and the variants of a fencepost algorithm described in class and in the book.

4) Given a simple method with parameters and a return value be able to describe in general terms what the method does.

5) Given variables (integers, doubles, Strings) along with values and a desired output format be able to provide a ***printf*** statement that will print the variable values given as defined.

6) Be familiar with the different types of input (from user, from file, from String) and sample code fragments for each.

7) Be able to evaluate compound logical expressions like the following (and remember short-circuiting):

```
( (4%2==1) || ( (3%2==1) && (9%2==1) ) )
```

8) Given code fragments with loops that produce output, be able to trace the code and indicate what the output would be.

For example, what would the following print to the console:

```
String alpha = "reascwefrq";
String output = "";
for (int i = 0; i < 10; i++) {
     if (i%2 == 0) {
          output += alpha.charAt(i);
     }
}
System.out.println(output);
```

9) Be able to write your own code fragment that uses a fencepost algorithm for a coding problem that is well suited for one.

10) Be able to write a cumulative sum, min loop, max loop, and sentinel loop that ask the user to supply entries.

11) Be prepared to develop a method or a small, complete program that reads text from a file (token-based and line-based).