CS200 Programming Assignment Grading Policy

January 25, 2009

1 Compiles and Runs: 30%

1. If the main program compiles and runs without crashing, students will receive full credit for this part.

2. If the program does not compile, the grader will see if it is a simple error (at most a one-line change, mis-capitalization or mis-spelling). If so, the grader will make the change and deduct 10 points from this portion of the grade.

3. If the program does not compile and there are multiple errors, students will lose the full 30 points along with points for correct output.

2 Correct Output: 30%

1. If the program produces the correct output students will receive full credit.

2. If the program produces some correct output but crashes, students will lose points.

3. If the program produces incorrect output, students will lose points.

4. If the program does not compile and run, students will lose all 30 points.

3 Documentation/Comments: 20%

1. All program files must include comments; undocumented code will lose the full 20 points.

2. The top of the class containing the main method should contain the following information: student name, ename, CS200 assignment number.

3. Each method should include a comment that describes how it works. For example, for a method writeToFile(String s), your comment might be:

   /**
    * writeToFile:
    *   This method writes the input string 's' to an output file.
    */
4. Include comments that clarify how code works. For example:

```java
// Find the maximum value in the integer array
int maxValue = 0;
for (int i=0; i<intArray.length; i++) {
    if (intArray[i] > maxValue)
        maxValue = intArray[i];
}
```

5. Using JavaDoc-style comments is optional, but will earn you extra-credit if you include the generated javadoc documentation in your submitted tarball. Here’s an example:

```java
/**
 * getLongestString:
 * This method returns the longest string in a string array.
 *
 * @param stringArray array of strings to scan.
 * @return longest string in the stringArray.
 * @throws IllegalArgumentException if ...
 **/```

## 4 Implementation and Coding Style: 20%

1. By default, the Eclipse editor will provide you with a clean, readable coding style.

2. The TA may deduct points for redundant code or unnecessary code. For example, in the code below, lines 3-4 and line 5 accomplish the same thing. Line 5 will never do anything useful:

```java
int maxValue = 0;
for (int i=0; i<intArray.length; i++) {
    if (intArray[i] > maxValue)
        maxValue = intArray[i];
    maxValue = Math.max(maxValue, intArray[i]);
}
```

3. Using meaningful names for variables and methods. For example, using `outputStream` for an output stream is more descriptive than `os` or `foo`.

4. Among the programs that produce correct output, the one that runs fastest will receive 10 pts extra-credit.