

# CS160 Programming Assignment Grading Policy

September 8, 2008

## 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. At a minimum, students should include the following comments at the top of every Java file:

```
// Name: student's name  
// Date: assignment due date  
// CS160 Assignment assignment number
```

3. Students may earn up to 5 extra credit points for:

- (a) Using comments to describe how each method 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.
 */
```

- (b) Including comments that clarify how code works. For example:

```
// 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];
}
```

## 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:

```
1 int maxValue = 0;
2 for (int i=0; i<intArray.length; i++) {
3     if (intArray[i] > maxValue)
4         maxValue = intArray[i];
5     maxValue = Math.max(maxValue, intArray[i]);
6 }
```

3. Students may earn up to 5 extra credit points for using meaningful names for variables and methods. For example, using `outputStream` for an output stream is more descriptive than `os` or `foo`.