Grading Rubrics for Assignment 10

CIS210 – Fall 2016
Point Value: 100 points
Assignment Due Date: 11:59 pm Thursday November 29, 2016

The students need to write a Java program, compile and run using Eclipse.

To receive full credit, students have to:
- Write a Java program (including comments) in the Eclipse editor to do exactly what the problem statement states.
- Type in appropriate identifying information in the header
- Compile and run the program
- Submit the source file correctly before the deadline

Turned in "on time" if received on Blackboard by 11:59 p.m. Thursday 11/29/2016
Turned in "late" if received on Blackboard by 11:59 p.m. Thursday 12/01/2016

Program Style – 20 points

Header identifies the student, course, assignment (Deduct up to 2 points)
- Deduct **2 points** each for not including student's name, or sec#
- Deduct **2 points** each for not including course, date, program number in the header, or instructor's name. (8 points)

Header describes program, input, output, assumptions (Deduct up to 2 points)
- Deduct **2 points** each for omitting descriptions of summary, input, output, or assumptions
- Deduct **2 points** for each section that is shortened to the point that it inadequately describes the program

In-line comments describe flow of program (Deduct up to 4 points)
- Deduct **2 points** for rude/ inappropriate comments
- Deduct **2 points** if critical or important internal comments are missing at less than **three** places.
- Deduct **2 points** if you are pointing out a pattern of deficient inline comments.
- Deduct **4 points** if all internal comments are missing or seriously lacking.

Vertical and horizontal spacing enhance readability (Deduct up to 4 points)
- Deduct **4 points** for failing to indent statements to make blocks and dependencies apparent
- Deduct **3 points** for not inserting blank lines to set off sections of the program/comments
- Deduct **3 points** for not aligning inline comments (also deduct if comments missing)

Incorrect Use of Constants (Deduct up to 2 points)
- Deduct **1 point** for each identifier that is not named appropriately.
- Deduct **1 point for each** incorrect identifier data type.
- Deduct **2 point** each for not declaring any constants.

Output well formatted and easy to read   (Deduct up to 6 points)
- Deduct **2 points** for rude/inappropriate messages.
- Deduct **3 points** if output is not properly spaced (horizontally or vertically)**.**
- Deduct **2 points** for missing or inappropriate introductory message and/or closing message.
- Deduct **2 points** for each number that is not well formatted.
- Deduct **2 points** if more than three of the output line widths exceed 80.

Program Coding & Output – (80 points)

Contains syntax warnings (if relevant) (Deduct up to 10 points)

- Deduct **2 points** each for syntax warnings that were introduced by the student other than those that lead to a runtime error (see below)

Does not run (has runtime or link errors) (Deduct up to 20 points)
- Deduct **20 points** if the student introduced a runtime error that causes program crash.

Gives incorrect results (has semantic errors) (Deduct up to 30 points)
- Deduct **5 points** for not displaying a warning message when the input file cannot be open and terminate the program gracefully.
- Deduct **5 points** for each incorrect calculation.
- Deduct **5 points** if they include unnecessary Java libraries and classes
- Deduct **5 points** if they miss necessary Java libraries and classes.
- Deduct **10 points** if the input data is not read from a file correctly.
- Deduct **10 points** if the output is not written to a file correctly.
- Deduct **4 points** for each incorrect result.
- Deduct **5 points** for incorrectly sorting the array in an ascending order of Employee ID.
- Deduct **5 points** for not writing output to a file named "employeeOutput.txt"

Use of Functions – (Deduct up to 20 points)
- Deduct **2 points** for each error that occurs inside a method
- Deduct **5 points** for not having a SelectionSort() method
- Deduct **5 points** for not having a ReadInputData() method
- Deduct **5 points** for not having a CalculateTotalCompensation() method in the *EmployeeCompensation* Interface
- Deduct **5 points** for not having a PrintManagerDividend() method.
- Deduct **5 points** for not having toString() method in Employee, JuniorEmployee SeniorEmployee, Manager classes.

Other Penalties

Incorrect File (98 points)
- Deduct **98 points** if the file is empty or not Java code.

No effort   (95 points)
- Deduct **95 points** if an attempt at the program was made, but there is no resemblance to a correct solution.

Late penalty (20 points)
- Deduct **20 points** if the project is turned in late.

Does NOT compile (50 points)
- Deduct **5 points** each for the first three compiler errors.
- Deduct **50 points** if program does not compile after 3 syntax error fixes.

Misnamed File (5 points)
- Deduct **5 points** if required file is turned in but is not named with correct name and extension

No effort regarding functions (40 points)
- Deduct **40 points** if no functions are used

Generic Penalty (15 points)
- Deduct **2 to 5 points** for each error found in program not specified in guidelines

Automatic **zeros**
- Program is identical to another student's program.
- Required file missing.