CST8116 Lab Exercise 06

Decisions

# Instructions Part 1: Pseudocode, Flowchart, Decisions

* A school wants a software program that permits a course number to be entered, and a letter grade. The output from the program is the course number, the percent grade range, the letter grade, and the numeric grade. One of the courses at the school, ABC333, has a requirement that if a student obtains a grade of D it is treated as an F. The program should handle this situation when generating the output.
* The grading system is summarized here:

Percent Grade Letter Grade Numeric Grade

80 – 100 A 4.0

70 – 79 B 3.0

60 – 69 C 2.0

50 – 59 D 1.0

0 – 49 F 0.0

* A junior programmer at the company you are working for has provided some starter code, a flowchart for method main, and UML Class diagrams you can use as a reference, however method createReport of class GradingSystem is incomplete. Focus your efforts on this method. (See Appendix for diagrams)
* Review the starter design and starter code provided and write pseudocode and create a flowchart for the createReport method of class GradingSystem. (You will need to demonstrate the use of both an if-statement, and a switch-statement within the method in solving this problem.)
* Hand drawn flowcharts are acceptable as images within the MS Word document, computer generated flowcharts can also be included in your MS Word document as an image, or you can use the MS Word shape(s) to create your flowchart. In all cases your full name must appear within the flowchart image either hand-drawn, or within a rectangle shape within the image or MS Word shape in the flowchart.

# Instructions Part 2: Testing

* Test the program with each valid input value, and one invalid input of a letter grade.
* This is an example, you will have more test cases.

|  |  |  |  |
| --- | --- | --- | --- |
| External Input (user) | Expected Output | Actual Output | Description |
| “SomeCourse”  A | Course: SomeCourse,  Letter Grade: A,  Percent: 80 to 100,  Number Grade: 4.0 | Course: SomeCourse,  Letter Grade: A,  Percent: 80 to 100,  Number Grade: 4.0 | Matches, algorithm is not required to verify course number. |
| “ABC333”  D | Course: ABC333  Letter Grade Entered: D,  Letter Grade Actual: F,  Percent: 90 to 100,  Number Grade: 4.0 | Course: ABC333  Letter Grade Entered: D,  Letter Grade Actual: F,  Percent: 90 to 100,  Number Grade: 4.0 | Matches, algorithm handles case of ABC333. |

# Instructions Part 3 Plan, Create, Compile, Run a Java Program using Decisions

* Using Eclipse, create a project named Exercise06 within your CST8116Workspace folder. (See Hybrid 1)
* You should be able to copy the starter .java files provided, then right-click on the src folder in your Eclipse project and select paste. The files will get placed into your src folder on the hard drive and you can edit them in Eclipse from that point.
* Follow Java programming conventions for identifiers, indentation, and provide programmer comments.
* Your implementation of your pseudocode must demonstrate both an if-statement and a switch-statement in the same program.xz
* Take screen shots of your program running, minimally three screen shots based on your test plan:
  + one with an invalid input for a letter grade;
  + one with ABC333 with D, and;
  + one with any other course and any valid letter grade
* Ensure your full name is visible as part of the program output in each screen shot.
* Provide programmer comments at the top of your source code file, and follow Java programming conventions for indentation and identifiers.

# Microsoft Word Document Format

See the template example (from exercise 01), suggested headings below:

Part 1: Pseudocode and Flowchart

Part 2: Test Plan

Part 3: Java Screen Shots

Note: You are not required to copy and paste your Java code into the MS Word document, however you must upload the .java file(s) in addition to your MS Word document.

# Submission Requirements

* Upload your MS Word document as well as your Java file to the Brightspace submission area by the due date. (See Brightspace for due date).
* Follow any additional submission requirements specified by your lab professor when submitting your homework.

# Grading

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Missing / Poor (0) | Below Expectations (0.5) | Meets Expectations (1) |
| Algorithm: pseudocode | Missing or incorrect. | Pseudocode is included in MS Word document, however there are logic mistakes and or missing either if or switch structure(s). | Pseudocode is included in MS Word document, provides a working algorithm that meets the requirements. |
| Algorithm: flowchart | Missing or wrong program or student name missing from flowchart. | Flowchart is included in MS Word document, has correct format, and however there are logic mistakes and or missing either if or switch structure. | Pseudocode is included in MS Word document, has correct format, and provides a working algorithm that meets the requirements. |
| Test Plan | Missing | Test plan does not test the minimal requested items. | Test plan does test the minimal requested items, and also adds tests so that each possible letter grade is tested. |
| Screen shots running program | Missing or missing students name as part of the program output and / or as part of the program comments. | Screen shots show program execution with samples taken from test plans. | Screen shots show program execution with samples taken from test plans, each image has brief line of text stating the test case. |
| Java file(s): Comments | No programmer comments, and or no student name. | Not all parts are commented: Comment header block at top of each source code file, class and constructor(s), and method(s) each have brief comment. | All parts are commented: Comment header block at top of each source code file, class and constructor(s), and method(s) each have brief comment. |
| Java file(s): Logic | No changes to starter code logic. | Method createReport() modified to solve the problem and-or logic closely matches the student’s algorithm and flowchart. Logic and / or syntax errors. | Method createReport() modified to solve the problem. Logic closely matches the student’s algorithm and flowchart. Logic is sound, Java programming conventions are followed. |

# Appendix: Sample Program Run(s) (Each block of output is a separate program run)

* User input is indicated using **bold font with yellow highlighting**.

Please enter course number **tuna**

Please enter letter grade **fish**

Invalid letter grade entered

Program by Stanley Pieda

Please enter course number **ABC333**

Please enter letter grade **tuna**

Invalid letter grade entered

Program by Stanley Pieda

Please enter course number **12345**

Please enter letter grade **A**

Course 12345: Entered Grade A, Percent Grade 80 - 100, Number Grade 4.0

Program by Stanley Pieda

Please enter course number **ABC333**

Please enter letter grade **D**

Course ABC333: Entered Grade D, Actual Grade F, Percent Grade 0 - 49, Number Grade 0.0

Program by Stanley Pieda

Please enter course number **ABC333**

Please enter letter grade **A**

Course ABC333: Entered Grade A, Percent Grade 80 - 100, Number Grade 4.0

Program by Stanley Pieda

Please enter course number **aaa**

Please enter letter grade **A**

Course aaa: Entered Grade A, Percent Grade 80 - 100, Number Grade 4.0

Program by Stanley Pieda

Please enter course number **aaa**

Please enter letter grade **B**

Course aaa: Entered Grade B, Percent Grade 70 - 79, Number Grade 3.0

Program by Stanley Pieda

Please enter course number **aaa**

Please enter letter grade **C**

Course aaa: Entered Grade C, Percent Grade 60 - 69, Number Grade 2.0

Program by Stanley Pieda

Please enter course number **aaa**

Please enter letter grade **D**

Course aaa: Entered Grade D, Percent Grade 50 - 59, Number Grade 1.0

Program by Stanley Pieda

Please enter course number **aaa**

Please enter letter grade **F**

Course aaa: Entered Grade F, Percent Grade 0 - 49, Number Grade 0.0

Program by Stanley Pieda

Please enter course number **ABC333**

Please enter letter grade **F**

Course ABC333: Entered Grade F, Percent Grade 0 - 49, Number Grade 0.0

Program by Stanley Pieda

Please enter course number **aaa**

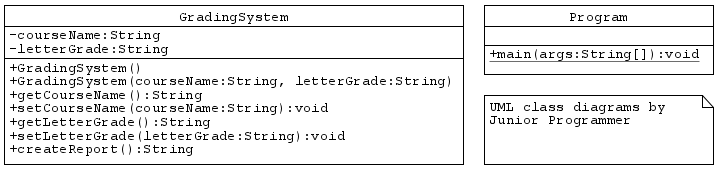
Please enter letter grade **a**

Invalid letter grade entered

Program by Stanley Pieda

# Appendix: Starter Diagrams

## UML Class Diagram (via UMLet)



## Flowchart (via Open Office Draw)

