CST8116 Lab Exercise 04 (21F)

# Instructions

* The five parts of the Software Development Process as presented by Cay Horstmann [1] will be used as the basis for this lab exercise.

1) Understand the problem

2) Develop and Describe an Algorithm

3) Test Algorithm with Simple Inputs

4) Translate the Algorithm into Java

5) Compile and Test Your Program

* Also refer to your textbook by Joyce Farrell [2] for help with pseucode, flowcharts, UML class diagrams.

# What is the problem to solve for this exercise?

* A movie theatre company would like a program that calculates the cost of an admission ticket, based on the age of the customer. (Ticket prices taken after [3], age ranges taken after [4]). The user will enter the age, then program output will be the price for a single ticket.
  + newborns, infants, toddlers zero (inclusive) to 2 years (exclusive) free admission
  + child, youth 2 to 18 years (both inclusive) $15.00
  + adult 19 (inclusive) to 65 years (exclusive) $20.00
  + senior 65 years or older (inclusive) $12.00
  + Any age entered that is less than zero, or greater than or equal to 130 should result in the program outputting an “invalid age entered” error message instead of a price.
* Ages are entered as non-decimal numbers.
* Output in the Java program should be via printf, with %f formatting output to 2 decimal places.
* Your full student name also needs to be part of the output.
* Some starter code is provided, you must use the starter code and cannot make changes to the structure of the provided classes, you need to complete the missing logic in the indicated methods ticketPrice and main.

# Part 1 Understand the Problem

* Review the word problem, as well as look at the starter code provided.
* Outline how the main logic would flow from inputs, to processing, to output.
* Use boolean expressions to plan the process of ticket pricing above into program logic, e.g.

Example: age < 0 OR age >= 130 report error

and so on…

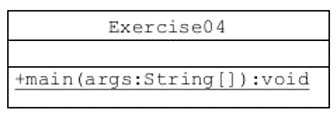
# Part 2a Pseudocode

* Create a detailed UML Class diagram that documents the provided class(s).
* Write pseudocode for the ticketPrice method
* Write pseucode for the main method
* Consider: Can a nested if structure be used to reduce the number of comparisons? See lecture notes week 9.

Reminder: Please review the lecture notes for UML Class diagrams as well as your Joyce Farrell textbook [2].

* In our course typically, we will have a class with only method main, as this is a static method it should be underlined.
* Local scope variables inside method main are not to be documented in the UML class diagram in the second row.
* Expanding on that, we never document variables that are local scope, i.e. declared within constructors or methods.
* The only variables that appear in a UML Class diagram, in the second row, are class-level variables declared at a class level.
* The only local-scope variables that appear in a UML class diagram, at all, are the parameters for constructors, and methods, but these are in-lined within each constructor’s or method’s line within the third row.

E.g.



Feel free to reproduce this example as part of your own UML Class diagrams.

## Part 2b Flowchart

* Write a flowchart for the ticketPrice() method
* Write a flowchart for the main method
* Place these into your MS Word document as images.
* Consider: Can a nested if structure be used to reduce the number of comparisons? See lecture notes week 9.

# Part 3 Test Plans for Algorithms

* Create a table in your MS Word document
* You can use the example below as a guide
* Refer to the lecture notes in week 9 regarding boundary testing, there are more than just two tests required.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Description |
| -1 | Invalid age entered | Invalid age entered | Matches, boundary check < 0 |
| 130 | Invalid age entered | Invalid age entered | Matches, boundary check >= 130 |
|  |  |  |  |

# Part 4 Translate the Algorithm into Java

* Create an Exercise04 project in Eclipse
* Using the starter code:
  + Update all of the comment sections as per the requirements of our course, see week 1 lecture notes
  + Follow your algorithm design and code the ticketPrice() method, as well as method main.

# Part 5 Compile and Test Your Program

* Run your program and take a screen shot of the running program, after it has completed a run.
* Re-use your algorithm test plan and test your program code, verifying that printf is correctly formatting to two decimal places.
* Test and document invalid and valid input like:
  + negative value for age and or age greater than 130 years
  + Boundary checks
  + What if the user entered “tuna” instead of an age?

# Microsoft Word Document Format

See the template example (from lab exercise 1) and use the suggested headings below:

Understand the Problem with Sample boolean expressions

Pseudocode(s) and Detailed UML Class Diagram

Flowchart(s)

Test Plan Algorithm

Screen Shot of Program Execution

Test Plan Program

# Submission Requirements

* You will need to submit your MS Word document and .java code files by the due date as specified in Brightspace.
* Follow your lab professor’s submission guidelines.

# Grading (8 Points)

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Missing / Incorrect (0) | Below Expectations (0.5) | Meets Expectations (1) |
| Understand the Problem | Missing or incorrect. | Partly correct. | Briefly outlines the necessary steps, in order, as an overview. Has examples of boolean expressions needed to solve the problem. |
| Pseudocode(s) | Missing or incorrect. | Partly correct. | Correct format, steps are in correct sequence and lead to correct outputs. Detailed UML class diagram is correct. Demonstrates selection structure use with nested if. |
| Flowchart(s) | Missing or incorrect. | Partly correct. | Correct format, correct shapes used, steps are in correct sequence, matching pseudocode and lead to correct outputs.  Demonstrates selection structure use with nested if. |
| Test Plan for Pseudocode and Flowchart | Missing or incorrect. | Partly correct. | Has correct format as shown in the lab handout, has test values and expected and actual outputs. Has coverage for boundary checks. |
| Screen shot executing program | Missing or incorrect or student name not part of program output. | Screen shot shows student name but program might not have compiled or run properly. | Screen shot shows student full name as in ACSIS. Program runs properly and valid input(s) and expected output(s) are shown. |
| Source Code: \*.java file(s) Comments and Conventions | Missing or poorly done. | Missing a comment-header from one or more of class declaration and / or method main declaration. Code loosely follows Java coding conventions for identifiers, indentation. | File comment header with student full name is present. Class and method declarations have comment headers. Code closely follows Java coding conventions for identifiers, indentation. This includes Java conventions for indentation of nested selection structure(s). |
| Source Code:  \*.java file(s) program structure and logic. | Missing or poorly done or program does not follow from the pseudocode, and flowchart(s). | Program may have small syntax mistakes and will not compile, and / or produces incorrect output when run. Program loosely follows the student’s pseudocode and flowchart(s). | Program has correct syntax and program logic that produces correct output. Program closely follows the student’s pseudocode and flowchart(s). Program logic for selection structure(s) is correct. |
| Test Plan for Program | Missing or poorly done or is only an unchanged copy of the provided algorithm test plan. | May not have correct format, does not verify that the program outputs match expectations. | Has correct format as shown in the lab handout, verifies that the program outputs match, and documents variations in output including samples of invalid inputs including invalid input that will crash the program. |

# Appendix: Sample Program Run

Enter age **17**

Price $15.00

Program by Stanley Pieda

Note that the user input in the example above was formatted with bold text and highlighted using MS Word formatting tools. Within the Eclipse IDE the default font color for user input is a light green.

# References

[1] Cay Horstmann. (2019). Big Java Early Objects. 7th Ed. Wiley.

[2] Joyce Farrell. (2018). Programming Logic & Design Comprehensive. 9th Ed. Cengage Learning.

[3] Movie Theatre Prices. (2020). Event Cinemas Prices. Last accessed on Oct 29, 2021. Retrieved from

<https://movietheaterprices.com/event-cinemas-prices/>

* This website goes into much more details regarding prices and options, keep to the simplified problem presented in the problem statement in this handout.

[4] Wikipedia. (). Child Development. Retrieved on Oct 29, 2021. Retrieved from <https://en.wikipedia.org/wiki/Child_development>