Assignment No. 4 Rubric

EECS 368 - Programming Language Paradigms

Due: 11:59 PM, Monday, October 17, 2022

**Student: Louis Tracy**

**Student ID: 3007878**

# Point Breakdown

|  |  |  |
| --- | --- | --- |
| ***Graded Value*** | ***Points Possible*** | ***Criteria*** |
|  | 3 | Name of the zip file: FirstnameLastname\_Assignment4 (with your first and last name) |
|  | 3 | Name of the Assignment folder within the zip file: FirstnameLastname\_Assignment4 |
|  | 3 | Copy of Rubric 4.docx with your name and ID filled out |
|  | 23 | Execution of PieChart.htm displays a pie chart with data set 1 showing correct customer response for each pie slice. |
|  | 46 | Execution of PieChart.htm displays a pie chart with data set 2 showing correct customer response for each pie slice. |
|  | 22 | PieChart.htm includes comments that adequately describe the code. |
|  | **100 pts** |  |

|  |  |  |
| --- | --- | --- |
| **Rubric for Program Comments\*** | | |
| **Exceeds Expectations**  **(90-100%)** | **Meets Expectations**  **(80-89%)** | **Unsatisfactory**  **(0-79%)** |
| Software is adequately commented with prologue comments, comments summarizing major blocks of code, and comments on every line. | Prologue comments are present but missing some items or some major blocks of code are not commented or there are inadequate comments on each line. | Prologue comments are missing all together or there are no comments on major blocks of code or there are very few comments on each line. |

Adequate Prologue Comments:

* Name of program contained in the file (e.g., EECS 368 Assignment 3)
* Brief description of the program, e.g.:
  + Simple tabbed interface using JavaScript and HTML
* Inputs (e.g., none, for a function, it would be the parameters passed to it)
* Output, e.g.,
  + Browser window with 3 tabs and text
* Author’s full name
* Creation date: The date you first create the file, i.e., the date you write this comment

Adequate comments summarizing major blocks of code and comments on every line:

* Provide comments that explain what each line of code is doing.
* You may comment each line of code (e.g., using //) and/or provide a multi-line comment (e.g., using /\* and \*/) that explains what a group of lines does.
* Multi-line comments should be detailed enough that it is clear what each line of code is doing.

# Grader Comments