Assignment No. 5
EECS 468
Programming Language Paradigms
Due: 11:59 PM, Wednesday, October 18, 2023
Submit deliverables in a single zip file to Canvas
Files in other formats (e.g., .tar) will not be graded
Name of the zip file: FirstnameLastname_Assignment5 (with your first and last name)
Name of the Assignment folder within the zip file: FirstnameLastname_Assignment5

Deliverables:

1. Copy of Rubric5.docx with your name and ID filled out (do not submit a PDF)
2. HTML file named: PieChart.htm (.html is also acceptable)
3. The grader will execute your HTML file to ensure it runs.

Assignment:

- In Chapter 17, we saw an example program that drew a pie chart.
- Modify this program so that the name of each category is shown next to the slice that represents it. (You must start with the code from the lecture and cannot use a charting module like "anychart").
- Try to find a pleasing-looking way to automatically position this text that would work for other data sets as well.
- You may assume that categories are big enough to leave ample room for their labels.
- You might need Math.sin and Math.cos again, which are described in Chapter 14.
- Draw the chart twice.
- Once with this customer satisfaction data (data set 1):
    - 104 of the customers answered "Very happy
    - 63 of the customers answered "Happy"
    - 87 of the customers answered "Neutral"
    - 51 of the customers answered "Unhappy"
    - 26 of the customers answered "Very unhappy"
    - 18 of the customers answered "No comment"
- Once with this customer satisfaction data (data set 2):
    - 927 of the customers answered "Satisfied"
    - 213 of the customers answered "Neutral"
    - 463 of the customers answered "Unsatisfied"
    - 510 of the customers answered "No comment"
- Label each chart as "Data Set 1" and "Data Set 2"
- You may choose your own colors for each pie slice
- Include all your code in a single HTML file named PieChart.htm
- Your code must display both pie charts when the grader clicks on PieChart.htm
- Provide comments for the JavaScript code that explain what each line of code is doing. See rubric below. You do not have to comment the HTML code.

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 468 Assignment 1)
- Brief description of the program, e.g.,
  - Hello World! examples 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 2 test buttons
- All collaborators
- Other sources for the code ChatGPT, stackOverflow, etc.
- 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.
- Each block of code must indicate whether you authored the code, you obtained it from one of the sources listed in the prolog, or one of your collaborators authored the code, or if it was a combination of all of these.

Collaboration and other sources for code:

- When you collaborate with other students or use other sources for the code (e.g., ChatGPT, stackOverflow):
  - Your comments must be significantly different from your collaborators.
  - More scrutiny will be applied to grading your comments in particular explaining the code "in your own words", not the source's comments (e.g., ChatGPT's comments).
- Failure to identify collaborators or other sources of code will not only result in a 0 on the assignment but will be considered an act of Academic Misconduct.
- Students who violate conduct policies will be subject to severe penalties, up through and including dismissal from the School of Engineering.