CS 153 / CS 453
Lab 2 - Strings and If Statements

- It is recommended that you keep your work organized in separate folders (directories).
- Before you begin, read Chapter 3, section 1 in the Zybook and read the documentation for Python string methods at https://www.w3schools.com/python/python_ref_string.asp W3Schools is a reputable web site for learning about Python and other languages.
- Add the following header comments to the top of all programs.

```
#  name of the file
#  CS 153 or CS 453
#  Written by ……
#  Date written ……
#  Date/time last modified  ……
#  Purpose:
#  Input:
#  Output:
```

Part A - strings.py (note: file names are case sensitive)

Write a Python program that will do the following:

1) Display a prompt and input a string from the user. The string may contain spaces.

Print meaningful messages along with your output.

2) Print the length of the string.

   Example input: `The sky is blue.`
   Correct output: `The length of the string is 16.`
   Incorrect output: `16`

3) Print the first word of the string. This must work for strings that have only one word as well as strings that have more than one word.
4) Print the string in all uppercase.
5) Print the string in all lowercase.
6) Print the string with all of the 'n' characters changed to 'm'.
7) Print the location of the first 'a' in the string.
8) Print the string with all of spaces removed.
9) Print the last character of the string.
10) If the string is longer than 8 characters, print the first 8 characters of the string. Otherwise, print the entire string.
11) Run the program several times. Each time, input a different string. Compare the **Actual** output to your **Expected** output. If there are errors, debug the program, compile again, and test again.

Part B - incomeTax.py (note: lowercase i and uppercase T)

Identifiers: There are two standard conventions for naming identifiers: underscore separator and camel case. Choose a convention that suits you and use it consistently.

Comments: All programs must have header comments and inline comments.

Income tax is calculated based on annual income. The tax rate is determined with a tiered approach. Income above a particular tier level is taxed at that level's rate.

1) Create a new Python program called incomeTax.py.

2) Input an integer value for the annual salary. (note: in all programs, choose a data type that is

3) Check to make sure that the salary is greater than or equal to zero. If not, print an error message and skip all of the calculations.

4) Determine the tax rate based on the tier in which the user's annual salary fits:

Tier	Salary Range	Tax Rate
1	0 - 20,000	10%
2	20,001 - 50,000	15%
3	50,001 - 100,000	20%
4	100,001 or higher	25%

5) Calculate the amount of tax owed. Round it to the nearest integer (nearest dollar).

6) Print output similar to the one shown below. Output must be aligned as shown.

```
Annual Salary:  $  15,000
Tax Rate:             10%
Tax Owed:       $   1,500
```

7) Run the program several times. Each time, input a different salary. Compare the **Actual** output to your **Expected** output. If there are errors, debug the program and test again.

Submit strings.py and incomeTax.py on Canvas.