# Lab Assignment Week 13

*CSC/DSCI 1301 – Principles of CS/DS I*

*Week of April 8th, 2024*

## Introduction

Welcome to the 13th programming lab of CSC/DSCI 1301! Today we will be covering the following topics:

- Creating User-Created Classes
- Defining Special Methods
- Defining Overloaded Operators
- Writing Exception Handlers
- Defining Custom Exceptions

### Lab policy reminders:

- Attendance is mandatory.
- Labs must be completed **individually.**
- TAs are here to help you. Ask them for help!
- Lab assignments are due at the end of each lab.

### Deliverables:

1. Python files for both programs.
2. Screenshots of the terminal output for the both programs.

If you have any questions, please do not hesitate to ask your TA!

## Program 1: Number.py

Write the class definition for a Custom User-Created **Number** Class. For this program, you will need to implement the Special and Operator methods within your own **Number** Class to mimic the functionality of the **Integer** Class. Your **Number** Class will store a single integer called **num** that is initialized within the **Number** Class constructor.

Your class must implement the following class customization methods to perform the following operations:

1. __str__(self): Returns a string of the value of **num** whenever an Object of your **Number** Class is used like a string.
2. __add__(self, other): Returns a new **Number** Object containing the sum of two **Number** Objects.
3. __sub__(self, other): Returns a new **Number** Object containing the subtraction of two **Number** Objects.
4. __mul__(self, other): Returns a new **Number** Object containing the multiplication of two **Number** Objects.
5. __truediv__(self, other): Returns a new Number Object containing the division of two **Number** Objects. Note: The value of **num** is always an **integer**.

Write a program that implements multiple objects of the **Number** class. Your program must demonstrate the functionality of all 5 specified Operators. Write 4 separate python expressions demonstrating each of the Operator Functions. Additionally write a 5$^{th}$ python expression that uses all 4 operators. After each arithmetic expression using the Number class. Write a Print statement that outputs the value of the Number class using the __str__ method.

### Example Output

```
25 + 5 = 30
25 - 5 = 20
25 * 5 = 125
25 / 5 = 5
( 30 + 20 * 5 ) / 125 = 1
```

### Skills Covered

- Creating User-Created Classes
- Defining Special Methods
- Defining Overloaded Operators

## Deliverables

For this program you will need to provide the python file containing your code as well as a screenshot of the output of your program. Please name your files as follows:

- Python Files
    - lastname_firstname_filename.py
    - For example: **hawamdeh_faris_number.py**
- Screenshots
    - lastname_firstname_filename.png
    - For example: **hawamdeh_faris_number.png**

## Program 2: step_counter.py

A pedometer treats walking 2,000 steps as walking 1 mile. Write a steps_to_miles() function that takes the number of steps as a parameter and returns the miles walked. The steps_to_miles() function raises a **ValueError** object with the message "Exception: Non-Numeric Value Entered." when the user input cannot be converted to an integer. The steps_to_miles() function raises a custom **NegativeValueError** object with the message "Exception: Negative Step Count Entered." when the number of steps is negative.

Write a program that reads the number of steps from a user, calls the steps_to_miles() function, and outputs the returned value from the steps_to_miles() function. Use a try-except block to catch any **ValueError**, or **NegativeValueError** object raised by the steps_to_miles() function and output the exception message. Output each floating-point value with two digits after the decimal point, which can be achieved as follows:

print(f'{your_value:.2f}')

### Example Output

```
Enter # of Steps:> 9623
4.81 miles

Enter # of Steps:> Five Hundred
Exception: Non-Numeric Value Entered

Enter # of Steps:> -1734
Exception: Negative Step Count Entered.
```

### Skills Covered

- Writing Exception Handlers
- Defining Custom Exceptions

## Deliverables

For this program you will need to provide the python file containing your code as well as a screenshot of the output of your program. Please name your files as follows:

- Python Files
    - lastname_firstname_filename.py
    - For example: **hawamdeh_faris_step_counter.py**
- Screenshots
    - lastname_firstname_filename.png
    - For example: **hawamdeh_faris_step_counter.png**