

# Homework 03

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

*Due Date: April 12<sup>th</sup> at midnight*

Please read all instructions, especially for the Program! Your answers to the short answer questions and program should be typed up. We will not accept handwritten or photocopied submissions.

## Classes

The **Vector** class consists of three attributes: **x**, **y**, and **z**. Each attribute represents the vector's magnitude in each of the three dimensions. The constructor of the **Vector** class takes in the three attributes as parameters.

1. Write the implementation of the `__add__` operator method, which returns a new Vector object with the new **x**, **y**, and **z** magnitudes.

## Exceptions

The following program reads integers `user_num` and `div_num` as input and calculates the quotient (`user_num` divided by `div_num`). It wants to execute the following Python Statements:

```
user_num = int(input(':> '))
div_num = int(input(':> '))
result = user_num / div_num
```

2. Write the try-except block surrounding the quotient to catch the `ValueError` and `ZeroDivision` exceptions. Each exception should have its own exception handler explaining the error to the user.

## Program: highest\_gpa.py

Implement a **Student** class that stores their first name, last name, and GPA. The **Student** class must contain three methods: `get_gpa()`, `get_first()`, `get_last()`. Each method will return the corresponding value stored within the **Student** object.

You will also need to implement a **Course** class. The **Course** class contains a single list called **roster**, which includes a list of **Student** objects that have been added to the **Course**. The **Course** class must implement the following methods: `add_student()` and `course_size()`.

- The `add_student()` method takes a **Student** object as a parameter and appends it to the **roster** list.
- The `course_size()` method returns the number of **Student** objects in the **roster** list.
- The **Course** class also needs to implement the `find_student_highest_gpa()` method.
  - This method returns the **Student** object of the student with the highest GPA in the course.

Write a program that allows a user to add multiple students to the course until a stop word has been entered. After the stop word, your program should print out the number of students enrolled in the course and the first, last, and gpa of the student with the highest GPA.

Your program should be able to handle the **ValueError** exception if a user enters a non-numeric GPA for a student. The program should not stop executing if a **ValueError** is raised. If the roster is empty, the `find_student_highest_gpa` method should raise a custom **EmptyRosterError** with the message 'Exception: Course Roster is Empty.'

## Example Output

### Normal Output

```
Welcome to CSC/DSCI 1301: Principles in CS/DS 1
Please Add Students to the Course: (quit or q to exit).
```

```
Enter First Name: Faris
Enter Last Name: Hawamdeh
Enter GPA: Three
Error: Enter a Numeric GPA
```

```
Enter First Name: Faris
Enter Last Name: Hawamdeh
Enter GPA: 3.0
```

```
Enter First Name: John
Enter Last Name: Smith
Enter GPA: 3.5
```

```
Enter First Name: Jane
Enter Last Name: Doe
Enter GPA: 3.9
```

```
Enter First Name: quit
```

```
Course Size: 3 students
Top Student: Jane Doe ( GPA: 3.9 )
```

### Empty Roster

```
Welcome to CSC/DSCI 1301: Principles in CS/DS 1
Please Add Students to the Course: (quit or q to exit).
```

```
Enter First Name: quit
```

```
Course Size: 0 students
Exception: Course Roster is Empty
```

## 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\_highest\_gpa.py**
- Screenshots
  - lastname\_firstname\_filename.png
  - For example: **hawamdeh\_faris\_highest\_gpa.png**

## Important

Your style of coding is just as important as the correctness of your program. Unlike your programs in the lab, you will be evaluated on your variable names and use of comments in addition to the program output. Your code must include a block comment at the top of your Python file with your name and a description of your program. Your variable names should be meaningful and conform to snake\_case.