GDAPS1 – Practice Exercise

Operator Overloading

# Objective

Practice overloading operators within two custom vector classes, ultimately making the vector objects easier to work with.

# Details

Start with a new **console application**, then create the following Vector2 and Vector3 classes:

## Vector2

* Fields & Properties
  + X and Y – doubles
* Constructors
  + Default – Set X and Y to zero
  + Parameterized – Accept values for X and Y
* Methods
  + Override ToString() to return a useful description of this vector

## Vector3

* Fields & Properties
  + X, Y and Z – doubles
* Constructors
  + Default – Set X, Y and Z to zero
  + Parameterized – Accept values for X, Y and Z
* Methods
  + Override ToString() to return a useful description of this vector

## Operator Overloads

Define the following operators for **each** of the classes inside of their appropriate classes:

* Vector + Vector
* Vector – Vector
* Vector \* scalar (a single number)
* Vector / scalar (a single number)

Also define the following implicit and explicit conversion operators:

* Implicit: Vector2 converted to Vector3 – Set the Z to zero
* Explicit: Vector3 converted to Vector2 – Drop the Z

## Main Method

In your main method, make at least two vectors of each type and **print them out**. Then do the following:

* Test out the operators you’ve declared by altering your vectors (add some together, multiply them by scalar values, etc.) Print out the results of the changes to ensure everything works.
* Next, create a List<Vector3> and store all of your vectors in that list. Yes, this includes storing your Vector2’s in that list - this will test implicit casting.
* Pull out an element and explicitly cast it to a Vector2. Print it out.
* Lastly, loop and average all of the vectors in the List, then print the final vector you end up with.

# Sample Run

Operator Overloading Test Run

--- Initial vectors ---

**(10, 10)**

**(2, 4)**

**(1.5, 2, 3.14159)**

**(5, 5, 5)**

--- After changes ---

Adding (10, 10) and (4, -13): **(14, -3)**

Multiplying (2, 4) by 2: **(4, 8)**

Subtracting (1, 2, 3) from (1.5, 2, 3.14159): **(0.5, 0, 0.14159)**

Dividing (5, 5, 5) by 2.5: **(2, 2, 2)**

--- After explicit cast ---

Element at index 3: **(2, 2)**

--- Average ---

**(5.125, 1.75, 0.5353975)**

# Submission

All of your work must be commented and follow this course’s coding standards. **Read through the Coding Standards document (located in MyCourses) to check over your code before you complete your program. Make sure you follow the coding standards for all code you create.**

1) Submit: Submit your program to the appropriate Assignments dropbox in MyCourses.

2) Check-off: Show your working program to the instructor or TA. If you do not finish before class ends, complete the exercise for homework and show one of us in-class on the next class period. If your program works as expected, you will be “checked off” to earn credit for the exercise.