**Topic 3: Structures**

Top of Form

Bottom of Form

**Content**

**Reading**

[Reading](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286811_1)

Structures

* + Microsoft:docs.microsoft.com
    - [Structure types](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/struct)
  + GeeksForGeeks: geeksforgeeks.com
    - [Structures](https://www.geeksforgeeks.org/c-sharp-structures-set-1/)
  + TutorialsPoint: tutorialspoint.com
    - [Structures](https://www.tutorialspoint.com/csharp/pdf/csharp_struct.pdf)

**Structures**

[Structures](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286811_1)

You learned about built-in data types (char, int, ...) Structures are a way to design your own storage containers in a language. Not all languages have a structure, in C# this functionality does exist!

public struct Address

{

public string houseNumber; // string not int to allow for 123B

public string streetName;

public string City;

public string State;

public string zip; // string not int to allow for 50000-000

}

Using the struct

Address home = new Address();

home.houseNumber = "123B";

home.streetName = "Main Street";

home.City = "Smalltown";

home.State = "Iowa";

home.zip = "50000";

How would you get the input from the user to the struct? Hmm... keep reading !

[**Student Structure**](https://dmacc.blackboard.com/webapps/assignment/uploadAssignment?content_id=_7286824_1&course_id=_102593_1&group_id=&mode=view)

Write a Console App that uses a structure that contains first name, last name,student id, major and year. Name your file StudentStructure.cs

* + Define a structure, Student that contains
    - first name
    - last name
    - student id
    - major
    - year in school (1, 2, ...)
  + Write a method DisplayStudent() that
    - Accepts a parameter of type Student
    - Returns a string
    - Is defined outside of the struct
  + Write a method CreateStudent()
    - Accepts parameters for each of the characteristics
    - Makes a new student of type Student
    - Returns an object of type Studen
    - Is defined outside of the struct
  + Add comments, including header (if you still need a reminder)
  + In your Main method
    - Prompt the user for input needed for one student
    - Call CreateStudent()
    - Call DisplayStudent()
  + Add Exception Handling and Input Validation

Submit StudentStructure.cs with Academic Honesty Header included.

This is 15 points