**Topic 1: Abstraction (OOP)**

Top of Form

Bottom of Form

**Content**

**Reading**

[Reading](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286729_1)

Object-Oriented Programming C# -- Abstraction

* + Microsoft:docs.microsoft.com
    - [How to define abstract properties (C# programming guide)](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/how-to-define-abstract-properties)
  + GeeksForGeeks: geeksforgeeks.com
    - [C# | Abstraction](https://www.geeksforgeeks.org/c-sharp-abstraction/)
  + TutorialsPoint: tutorialspoint.com
    - [C# - Abstract Classes in C#](https://www.tutorialspoint.com/Abstract-Classes-in-Chash)
  + W3Schools: w3schools.com
    - [C# Abstraction](https://www.w3schools.com/cs/cs_abstract.asp)

**Abstract Classes**

[Abstract Classes](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286729_1)

https://youtu.be/yHenEU03WXk

Abstract class serve as a base class that does not get instantiated, you cannot create object of an abstract class, you must derive the abstract class and override methods.

So what does all that mean?

From  [How to define abstract properties](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/how-to-define-abstract-properties)

[**Console App: Abstract Class**](https://dmacc.blackboard.com/webapps/assignment/uploadAssignment?content_id=_7286745_1&course_id=_102593_1&group_id=&mode=view)

Time to test your abstract class skills. You will create an abstract class Sport. All sports are played, but how they are played differs for each sport. This is a good candidate for an abstract method in a sports class. You may have heard "Go Sports!" but in reality, this class will not be made into an object, the sub classes will Create an abstract class called Sport

Properties (these are not private!)

* + Name- name of the sport, decide appropriate datatype
  + Equipment- needed to play the game
  + Number of players- on a team

Methods

* + Play()- returns a string describing the game
  + ToString()- Displays properties as below
  + Falcons requires 12 people, and Field, Goals and Rugby Ball.

Eagles requires 3 people, and Track Field and Shotput.

Define 2 classes, Rugby and ShotPut.

* + In Rugby
    - Define constructors
    - Override the abstract method of Sport, return one sentence desribing the play of the game
    - Add a method to FormTeam() that returns the number of players needed to form a team
  + In ShotPut
    - Define constructors
    - Override the abstract method of Sport, return one sentence desribing the play of the game

Driver

* + Make objects of both classes (this is shown below)
  + public static void Main(string[] args)
  + {
  + Rugby rugby = new Rugby("Falcons", "Rugby Ball", 12);
  + Sport shotput = new ShotPut("Eagles", "Field and Shot put", 3);
  + Console.WriteLine(rugby.ToString());
  + Console.WriteLine(shotput.ToString());
  + // add method calls as instructed here
  + }

​

* + Test Rubgy Class
    - Call Play()
    - Call FormTeam()
  + Test ShotPut Class
    - Call Play()
  + Use Console.WriteLine() to print results. DO NOT include Console.WriteLine() in your classes.

Extra Credit: Can you use Sport as the data type for Rugby? Why or why not?  Answer in the comment Box of the assignment for up to 2 extra points.

Submit Sport.cs, ShotPut.cs, Rugby.cs and Program.cs. Include your Academic Honesty Header only in Program.cs

This is worth 20 points