Lab 6

# Objectives

Practice C# code writing that involves arrays.

# Instructions

* Install Visual Studio 2017 or Visual Studio 2019 to your computer.<https://visualstudio.microsoft.com/>
* For each question below submit your C # and also screenshots showing how your program compiles and executes (sample outputs)- You can upload your C# project but alternatively create a document with your C# code **text** copied and pasted and your screenshots pasted and then upload the document to Blackboard for submission. Prof. Aydin should be able to run your submitted code.
* Your name should appear on the screenshots for receiving full credit.
* **Academic Integrity:** If you are stuck when working in this lab you can collaborate with a couple of classmates. In that case, make sure to write/submit the name of your collaborators and any web site you used as a resource to understand the concepts and lab questions and to complete your code to prevent plagiarism and breach of academic integrity.
  + You are *not* allowed to directly copy code from the Internet, your friends, and other resources without spending any effort in completing the work. Make sure to review the academic integrity policy in the syllabus and ask for clarification, if needed.

1. Take a look at the program below.

* This program is incomplete. Your job is to update the classes and the main program to allow Baseball players to be sorted based on a criteria the user selects from a menu that you show to the user. The user may want to see the list of baseball players sorted alphabetically, from smallest to biggest salary, or from smallest number of hit to biggest hit statistics
* Hint:
  + Review [Week 8 (Chapter 7) slides](https://drive.google.com/drive/folders/1-voMzBRitOdXiBNl8SrMm_WM-CbnhxQD?usp=sharing) and see which one of **IComparable, IComparable<T>, IComparer, IComparer<T> interfaces** your classes in the program should implement to be able to use [Array.Sort](https://docs.microsoft.com/en-us/dotnet/api/system.array.sort?view=netcore-3.1) method in .NET
  + You can modify the given classes in the program and add a new class(es) as you find appropriate

using System;

namespace LearnCSharp

{

public abstract class Player

{

public string Name { get; set; }

public double Salary { get; set; }

public override string ToString() => $"{ Name } with salary of ${ Salary }";

public abstract void displayStatistics();

}

public class BaseballPlayer : Player

{

public int AtBats { get; set; }

public int HomeRuns { get; set; }

public int Hits { get; set; }

public override void displayStatistics()

{

Console.WriteLine($"At Bats: { AtBats } \nHome Runs: { HomeRuns }\n Hits: { Hits }");

}

}

class Program

{

static void Main()

{

// create a one dimensional array of baseball

// players with size of 100

//...

// add three baseball players to the array

//...

// show menu to the user

// 1. sort players based on salary and display

// 2. sort players based on hits and display

// 3. sort players based on name and display

// …

// ask the user what option they want from

// the menu and accordingly display all the information

// of the baseball players in that specific order

//...

}

}

}