Connor Perron
BCS426
Prof. Aydin
Spring 2021

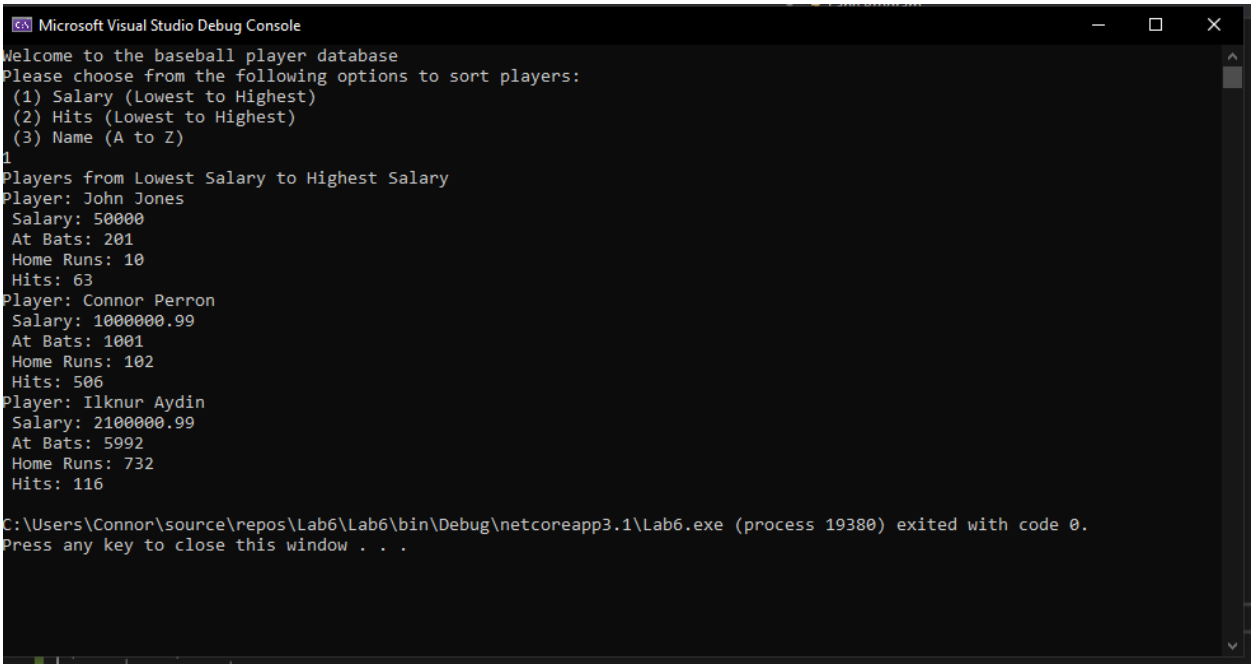
```
using System;
using System.Text;
using System.Collections;
using System.Collections.Generic;

namespace Lab6

{
    2 references
    public abstract class Player
    {
        private string _name;
        private double _salary;

        1 reference
        public Player (string name = "", double salary = 0.0)
        {
            _name = name;
            _salary = salary;
        }
        4 references
        public string Name { get; set; }

        4 references
        public double Salary { get; set; }

        3 references
        public override string ToString() => $"{ _name } with salary of ${ _salary }";

        4 references
        public abstract void displayStatistics();



    }
```

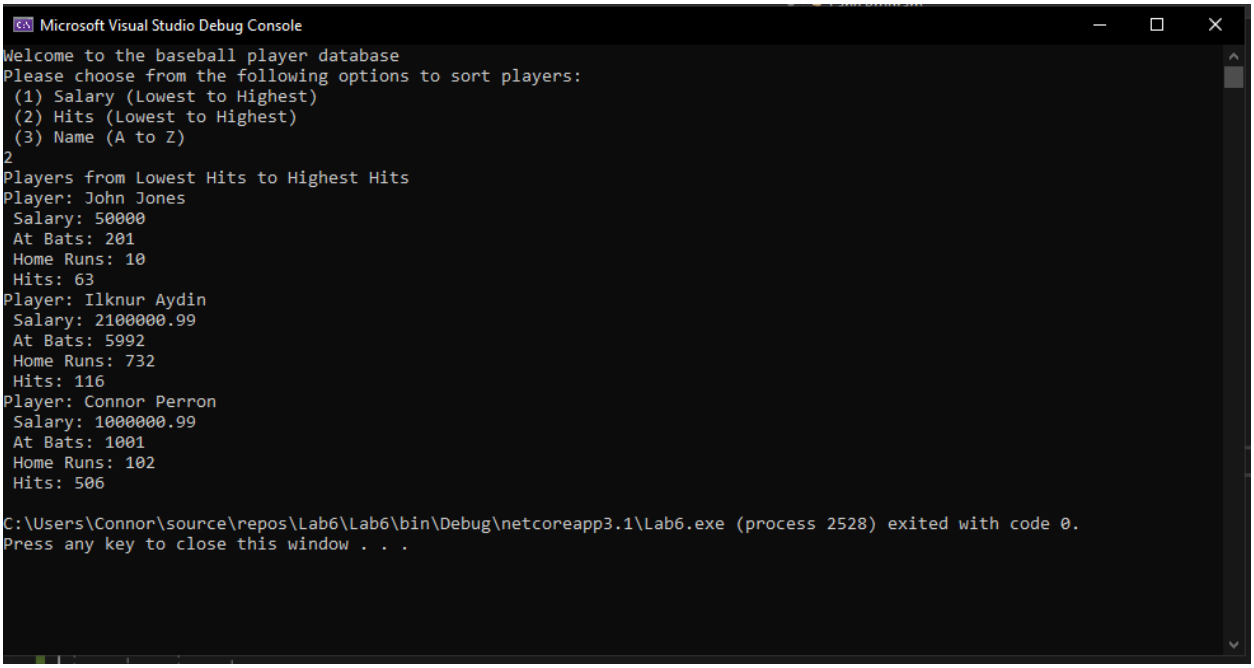
```
7 references
public class BaseballPlayer : Player
{
    private int _atBats;
    private int _homeRuns;
    private int _hits;

    4 references
    public BaseballPlayer(string name = "" , double salary = 0.0 , int atBats = 0 , int homeRuns = 0, int hits = 0): base(name, salary)
    {
        this.Name = name;
        this.Salary = salary;
        _atBats = atBats;
        _homeRuns = homeRuns;
        _hits = hits;
    }
    0 references
    public int AtBats { get; set; }

    0 references
    public int HomeRuns { get; set; }

    2 references
    public int Hits { get; set; }

    4 references
    public override void displayStatistics()

    {

        Console.WriteLine($"Player: {this.Name} \n Salary: {this.Salary} \n At Bats: { _atBats } \n Home Runs: {  _homeRuns }\n Hits: { _hits }");

    }



}
```

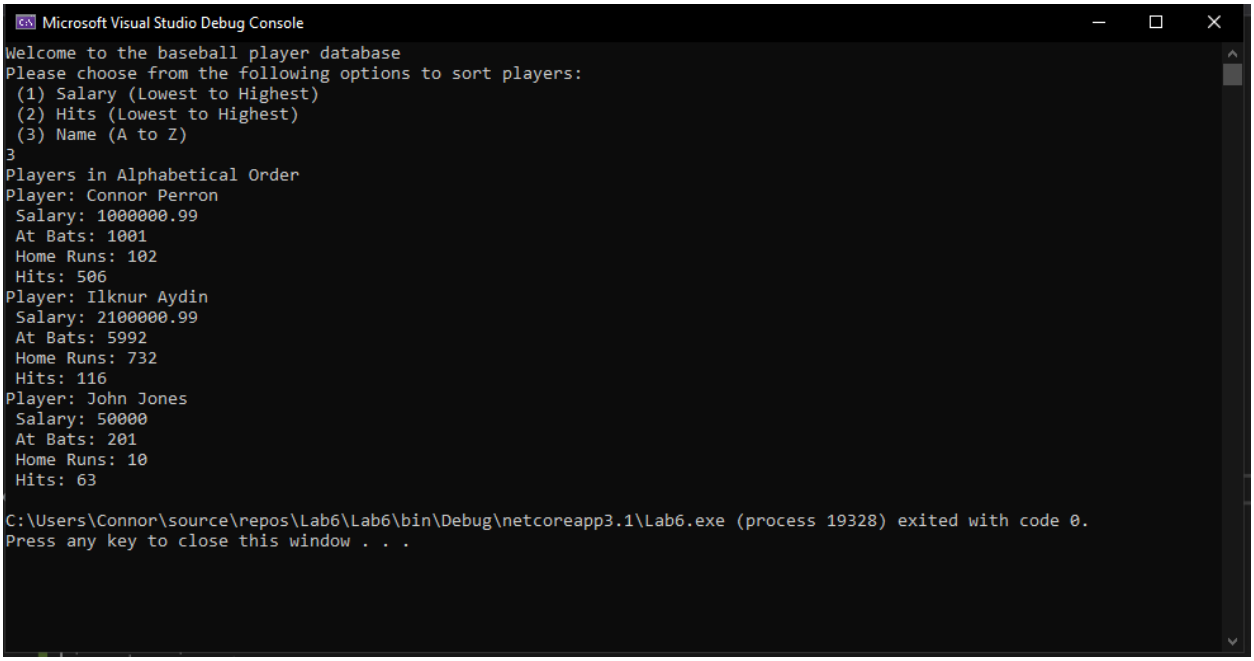
```
0 references
static void Main()
{

    // create a one dimensional array of baseball
    // players with size of 100
    BaseballPlayer[] team = new BaseballPlayer[100];

    // add three baseball players to the array
    BaseballPlayer b1 = new BaseballPlayer("Connor Perron", 1000000.99, 1001, 102, 506);
    BaseballPlayer b2 = new BaseballPlayer("Ilknur Aydin", 2100000.99, 5992, 732, 116);
    BaseballPlayer b3 = new BaseballPlayer("John Jones", 50000.00, 201, 10, 63);
    BaseballPlayer temp = new BaseballPlayer();
    team[0] = b1;
    team[1] = b2;
    team[2] = b3;
    // 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
    Console.WriteLine("Welcome to the baseball player database");
    Console.WriteLine("Please choose from the following options to sort players: \n (1) Salary (Lowest to Highest) \n (2) Hits (Lowest to Highest) \n (3) Name (A to Z)");

    // ask the user what option they want fromthe menu
    int choice = Convert.ToInt32(Console.ReadLine());
```

```
if (choice == 1)
{
    Console.WriteLine("Players from Lowest Salary to Highest Salary");
    for (int i =0; i < 2; i++)
    {
        if (team[i].Salary < team[i+1].Salary)
        {
            temp = team[i];
            team[i] = team[i + 1];
            team[i + 1] = temp;
        }

    }

    for (int j = 2; j >= 0; j--)
    {
        team[j].displayStatistics();
    }
}

else if (choice == 2)
{
    Console.WriteLine("Players from Lowest Hits to Highest Hits");
    for (int i = 0; i < 2; i++)
    {
        if (team[i].Hits < team[i + 1].Hits)
        {
            temp = team[i];
            team[i] = team[i + 1];
            team[i + 1] = temp;
        }

    }

    for (int j = 2; j >= 0; j--)
    {
        team[j].displayStatistics();
    }
}
```

```
        else if (choice == 3)
        {
            Console.WriteLine("Players in Alphabetical Order");
            for (int i = 0; i < 2; i++)
            {
                string name = team[i].Name;
                int compare = name.CompareTo(team[i + 1].Name);

                if (compare > 0)
                {
                    temp = team[i];
                    team[i] = team[i + 1];
                    team[i + 1] = temp;
                }
            }


            for (int j = 0; j < 3; j++)
            {
                team[j].displayStatistics();
            }
        }

        else
        {
            Console.WriteLine("Invalid Choice :(");
        }



    }

}
```


Microsoft Visual Studio Debug Console
Welcome to the baseball player database
Please choose from the following options to sort players:
(1) Salary (Lowest to Highest)
(2) Hits (Lowest to Highest)
(3) Name (A to Z)
1
Players from Lowest Salary to Highest Salary
Player: John Jones
Salary: 50000
At Bats: 201
Home Runs: 10
Hits: 63
Player: Connor Perron
Salary: 1000000.99
At Bats: 1001
Home Runs: 102
Hits: 506
Player: Ilknur Aydin
Salary: 2100000.99
At Bats: 5992
Home Runs: 732
Hits: 116
C:\Users\Connor\source\repos\Lab6\Lab6\bin\Debug\netcoreapp3.1\Lab6.exe (process 19380) exited with code 0.
Press any key to close this window . . .


Microsoft Visual Studio Debug Console
Welcome to the baseball player database
Please choose from the following options to sort players:
(1) Salary (Lowest to Highest)
(2) Hits (Lowest to Highest)
(3) Name (A to Z)
2
Players from Lowest Hits to Highest Hits
Player: John Jones
Salary: 50000
At Bats: 201
Home Runs: 10
Hits: 63
Player: Ilknur Aydin
Salary: 2100000.99
At Bats: 5992
Home Runs: 732
Hits: 116
Player: Connor Perron
Salary: 1000000.99
At Bats: 1001
Home Runs: 102
Hits: 506
C:\Users\Connor\source\repos\Lab6\Lab6\bin\Debug\netcoreapp3.1\Lab6.exe (process 2528) exited with code 0.
Press any key to close this window . . .


Microsoft Visual Studio Debug Console
Welcome to the baseball player database
Please choose from the following options to sort players:
(1) Salary (Lowest to Highest)
(2) Hits (Lowest to Highest)
(3) Name (A to Z)
3
Players in Alphabetical Order
Player: Connor Perron
Salary: 1000000.99
At Bats: 1001
Home Runs: 102
Hits: 506
Player: Ilknur Aydin
Salary: 2100000.99
At Bats: 5992
Home Runs: 732
Hits: 116
Player: John Jones
Salary: 50000
At Bats: 201
Home Runs: 10
Hits: 63
C:\Users\Connor\source\repos\Lab6\Lab6\bin\Debug\netcoreapp3.1\Lab6.exe (process 19328) exited with code 0.
Press any key to close this window . . .