FULL SAIL UNIVERSITY

COP2334

PG2 – LAB: HEROES V1

CONTENTS

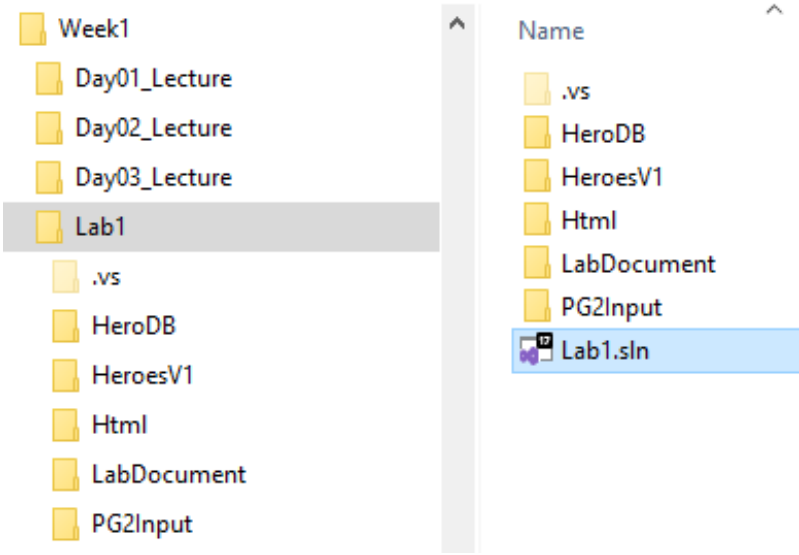
Setup 2
Lab Video 2
Part A 3
Lecture Videos for Part A 3
Tips: 3
Part A-1: ShowHeroes 4
Part A-2: PrintHero 4
Part A-3: FindHero 5
Part B 6
Lecture Videos for Part B 6
Tips: 6
Part B-1: RemoveHero 7
Part B-2: StartsWith 7
Part B-3: RemoveAllHeroes 8
Part C 9
Lecture Videos for Part C 9
Tips: 9
Part C-1: Optional parameter 9
Rubric 10
Part A 10
Part B 10
Part C 10

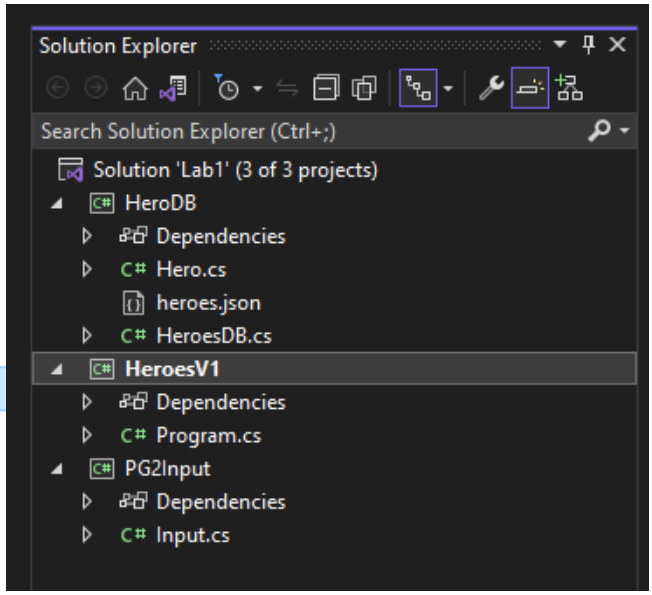
FULL SAIL UNIVERSITY

COP2334

SETUP

A **C# .NET Core console application** has been provided for you in your **GitHub repo**. **Use the provided solution.**





Lab Video

Here's a video showing what the lab could look like when completed:

https://web.microsoftstream.com/video/3ed0e560-a731-4e00-bb94-21abb6551b0b

FULL SAIL UNIVERSITY

COP2334

PART A

Lecture Videos for Part A

METHODS LECTURE:

https://fullsailedu-my.sharepoint.com/:v:/g/personal/ggirod_fullsail_com/EW0hLKhQiBdFjOGq1WG6oRoB9TTQWJd1L9ic6VRiEYwgdg?e=J7uZXt

Chapters: **Method Basics** through **Method Examples**.

LIST LECTURE:

https://fullsailedu-my.sharepoint.com/:v:/g/personal/ggirod_fullsail_com/ERG1iZHKJgFIoj6W8dhxPPgBIIY-Ot1b6Ueh-50Ggfcikg?e=goT9d6

Chapters: **Array Basics** through **Looping Examples**.

Tips:

HOW TO WORK WITH THE CONSOLE:

Console Class in C# with Examples - Dot Net Tutorials

Console Lecture

ACCESSING THE PARTS OF THE HERO CLASS:

If you have a Hero object, then to access any of the elements inside the object, you use the **dot operator**.

Example to show the Speed of the hero…

```
Hero myHero = _heroes[0];  //get the first hero in the list

Console.WriteLine($"The speed of the hero: {myHero.Powerstats.Speed}");  //use the dot operator to access the Speed value of the Powerstats property of the myHero variable.
```

METHODS

Creating and callings methods: C# Methods (w3schools.com)

Parameters and arguments: C# Method Parameters (w3schools.com)

Using returned values: C# Return Values (w3schools.com)

LIST<T>

Working with the List class: C# List Tutorial (c-sharpcorner.com)

Part A-1: ShowHeroes

Add a method called **ShowHeroes** to the **HeroesDB** class which is in the HeroDB project (see the screenshot in the Setup section above)**.** The method should loop over the _heroes list and print the hero ID and the hero Name.

NAME	RETURNS	PARAMETERS	COMMENTS
ShowHeroes	nothing	nothing	Show the ID and name for each hero.

In **Main** (which is in **Program.cs** in the HeroesV1 project), add code to case 1 of the switch to call the method.

```
  1: A-Bomb
  2: Abe Sapien
  3: Abin Sur
  4: Abomination
  5: Abraxas
  6: Absorbing Man
  7: Adam Monroe
  8: Adam Strange
 10: Agent Bob
 11: Agent Zero
 12: Air-Walker
 13: Ajax
 14: Alan Scott
 15: Alex Mercer
 17: Alfred Pennyworth
 18: Alien
 20: Amazo
 23: Angel
 24: Angel-Warren
 25: Angel Dust
 26: Angel Salvadore
 28: Animal Man
 29: Annihilus
 30: Ant-Man
 31: Ant-Man II
 32: Anti-Monitor
```

Part A-2: PrintHero

Add a method called **PrintHero** to the **HeroesDB** class. The method should have a Hero parameter passed to it. Print the details of the Hero parameter. (see the screenshot in part A-3) **Make sure to match the formatting** (color and indention).

NAME	RETURNS	PARAMETERS	COMMENTS
PrintHero	nothing	Hero	Show the details of the Hero parameter.

You will call PrintHero in part A-3. That's when you'll see if it works correctly or not.

Part A-3: FindHero

Add a method called **FindHero** to the **HeroesDB** class. The method should have a string parameter for the name to search. The method needs to loop over the heroes list to try to find the hero. Check if each hero name matches the parameter. If so, break out of the loop and return the found hero. When checking for a match, **make sure to ignore the case** of the name and the parameter.

NAME	RETURNS	PARAMETERS	COMMENTS
FindHero	Hero	string	Returns a Hero if found otherwise returns null.

In Main (which is in Program.cs in the HeroesV1 project), add code to case 2 of the switch. Using **Input.GetString**, ask the user to enter the name of the hero to find. Call **FindHero** passing the string that the user enters and store the result in a Hero variable. In Main, if the returned value from FindHero is not null, then call **PrintHero** else print out a message that the name was not found.

```
Please enter the name of the hero to find: Batman


69: Batman
        STATS:
                Intelligence: 81
                Strength: 40
                Speed: 29
                Durability: 55
                Power: 63
                Combat: 90
        APPEARANCE:
                Race: Human
                Height: 5'10
                Weight: 170 lb
                Eye Color: Blue
                Hair Color: Black
        BIO:
                Full Name: Terry McGinnis
                Alter Egos: No alter egos found.
                Aliases: Batman II,The Tomorrow Knight,
                Place Of Birth: Gotham City, 25th Centu
                First Appearance: Batman Beyond #1
                Publisher: DC Comics
                Alignment: good
        WORK:
                Occupation: -
        CONNECTIONS:
                Group Affiliation: Batman Family, Justi
                Relatives: Bruce Wayne (biological fath
```

```
Please enter the name of the hero to find: Bob
Bob was not found.
```

FULL SAIL UNIVERSITY

COP2334

PART B

Lecture Videos for Part B

METHODS LECTURE:

https://fullsailedu-my.sharepoint.com/:v:/g/personal/ggirod_fullsail_com/EW0hLKhQiBdFjOGq1WG6oRoB9TTQWJd1L9ic6VRiEYwgdg?e=J7uZXt

Chapters: **Parameters by Value** through **Out Parameters Examples**.

LIST LECTURE:

https://fullsailedu-my.sharepoint.com/:v:/g/personal/ggirod_fullsail_com/ERG1iZHKJgFIoj6W8dhxPPgBIIY-Ot1b6Ueh-50Ggfcikg?e=goT9d6

Chapters: **Removing Items** through **Removing Items Examples**.

Tips:

Ref parameters: C# - Passing Parameters by Reference (tutorialspoint.com)

Out parameters: C# - Passing Parameters by Output (tutorialspoint.com)

Removing from a List:

List<T>.Remove(T) Method

List<T>.RemoveAt(Int32) Method

FULL SAIL UNIVERSITY

Part B-1: RemoveHero

Add a method called **RemoveHero** to the **HeroesDB** class. The method should have a string parameter for the name of the hero to remove. The method should loop over the heroes list. If the hero is found, remove the hero from the heroes list. Return true if the hero was found and removed. Return false if the hero was not found. When checking for a match, **make sure to ignore the case** of the name and the parameter.

NAME	RETURNS	PARAMETERS	COMMENTS
RemoveHero	bool	string	Removes the hero from the heroes list if it is found and returns true. If not found, returns false.

In Main, add code to case 3 of the switch. Using **Input.GetString**, ask the user to enter the name of the hero to remove. Call **RemoveHero** passing the string that the user enters. In Main, if the returned value is true, print that the hero was removed else print that the hero was not found.

```
Please enter the name of the hero to remove: Aquaman
Aquaman was removed.
```

```
Please enter the name of the hero to remove: Bob
Bob was not found.
```

Part B-2: StartsWith

Add a method called **StartsWith** to the **HeroesDB** class. The method should have a string parameter for the name of the hero to match and a **ref parameter** for the List of heroes that were found. Loop over the heroes list and add every hero whose name **starts with** the string parameter to the List parameter. When checking for a match, **make sure to ignore the case** of the name and the parameter.

NAME	RETURNS	PARAMETERS	COMMENTS
StartsWith	nothing	string ref List<Hero>	Finds every hero whose name starts with the string parameter. Add the heroes to the list parameter.

In Main, add code to case 4 of the switch. Using **Input.GetString**, ask the user to enter the first part of the names to find. Call **StartsWith** passing the string that the user enters and a ref parameter for the list. Make sure to create an empty list. In Main, print out the number of heroes found AND loop over the list and call **PrintHero** for each hero found.

For example, if the user types Bat, the method should find all of the heroes whose name starts with Bat.

```
Please enter the start of the name to find: Aqua
Found 3 heroes that start with Aqua


36: Aquababy
        STATS:
                Intelligence: 10
                Strength: 16
                Speed: 12
                Durability: 14
                Power: 37
```

Part B-3: RemoveAllHeroes

Add a method called **RemoveAllHeroes** to the **HeroesDB** class. The method should have a string parameter for the name of the hero to match and an **out parameter** for the List of heroes that were found and removed. Loop over the heroes list and add every hero whose name **starts with** the string parameter to the List parameter. Make sure to remove the hero from the heroes list.

Be sure to initialize the list inside the RemoveAllHeroes method.

NAME	RETURNS	PARAMETERS	COMMENTS
RemoveAllHeroes	nothing	string out List<Hero>	Finds every hero whose name starts with the string parameter and removes the hero. Add the removed heroes to the list parameter.

In Main, add code to case 5 of the switch. Using **Input.GetString**, ask the user to enter the first part of the names to remove. Call **RemoveAllHeroes** passing the string that the user enters and an out parameter for the list. Because it's an out parameter, there is no need to initialize the out parameter before calling RemoveAllHeroes. In Main, after calling the method, if the list is empty, print that "No heroes found that start with <the startsWith string the user entered>" else print "The following heroes were removed: " and loop over the list calling **PrintHero** for each hero in the list.

```
Please enter the start of the name of the heroes to remove: Aqua
The following heroes were removed:


38: Aquaman
        STATS:
                Intelligence: 81
                Strength: 85
                Speed: 79
                Durability: 80
```

```
Please enter the start of the name of the heroes to remove: Steve
No heroes found that start with Steve.
```

FULL SAIL UNIVERSITY

COP2334

PART C

Lecture Videos for Part C

METHODS LECTURE:

https://fullsailedu-my.sharepoint.com/:v:/g/personal/ggirod_fullsail_com/EW0hLKhQiBdFjOGq1WG6oRoB9TTQWJd1L9ic6VRiEYwgdg?e=J7uZXt

Chapters: **Optional Parameters** through **Optional Parameters Examples**.

LIST LECTURE:

https://fullsailedu-my.sharepoint.com/:v:/g/personal/ggirod_fullsail_com/ERG1iZHKJgFIoj6W8dhxPPgBIIY-Ot1b6Ueh-50Ggfcikg?e=goT9d6

Chapters: **Arrays to Lists** through **Cloning Lists Examples**.

Tips:

Optional parameters: C# Default Parameter Value (Optional Parameter) (w3schools.com)

Part C-1: Optional parameter

Add an **optional parameter** to the ShowHeroes method. Default it to 0. In the method, if the parameter has the default value, show all the heroes else show the number of heroes that match the parameter. Example, if 10 is passed in, only show the first 10 heroes.

In Main, add code to case 6 of the switch. Using **Input.GetInteger**, ask the user to enter the number of heroes to show. Use the **HeroesDB.Count** property to get the max value to pass to **Input.GetInteger**. Call **ShowHeroes** and pass in the number that **Input.GetInteger** returns.

```
How many heroes to show? 10
  1: A-Bomb
  2: Abe Sapien
  3: Abin Sur
  4: Abomination
  5: Abraxas
  6: Absorbing Man
  7: Adam Monroe
  8: Adam Strange
 10: Agent Bob
 11: Agent Zero
```

FULL SAIL UNIVERSITY

COP2334

RUBRIC

Part A

FEATURE	POINTS
Part A-1: ShowHeroes	15
Part A-2: PrintHero	15
Part A-3: FindHero	15
TOTAL	45

Part B

FEATURE	POINTS
Part B-1: RemoveHero	15
Part B-2: StartsWith	15
Part B-3: RemoveAllHeroes	15
TOTAL	45

Part C

FEATURE	POINTS
Part C-1: Optional Parameter	10
TOTAL	10