WRAV101/MSEV101: Practical 7 (Tuesday 15:45)

Show your completed tasks to the assistant before the end of the session.

Objectives

- Creating and using methods

How do I submit my work?

Show your completed tasks to an assistant on duty **before the end** of the session. Your submissions will be assessed using the rubric provided at the end of this document.

Note: Remember to test your programs with a variety of values.

Task 1 *(*Create a new project *(Console App (.Net Framework))* and name it *P7T1)*

Create a user defined method called `GetTemp()` that takes no input (as an argument) and returns a valid temperature value between `-10` and `45` (a temperature value can have decimal places). If the user enters an invalid value, the method should repeatedly prompt the user for a value until a valid value is entered. When the method has a valid value, it should return that value to the calling program.

Write a user defined method named `IsCold` that can be used to determine whether a temperature can be classified as cold (less than `18`). The method must take as input a temperature, and return a `boolean`, indicating whether the temperature is classified as cold.

Write a program that reads in `5` temperatures by using the `GetTemp` method to ensure that valid temperatures are processed. You need to calculate the average of the temperatures. Your program must also use the `IsCold` method to determine the number of cold temperatures entered. When the `5` temperatures have been processed you need to display the average and number of cold temperatures (from the temperatures entered).

```
Enter a temperature: 25
Enter a temperature: 28
Enter a temperature: 46
46 is not a valid temperature, enter a value between -10 and 45.
Enter a temperature: 30
Enter a temperature: 18
Enter a temperature: 17
The average of the temperatures: 23.6
The number of cold days        : 1
```

Now update your program to calculate the average temperature for each week (comparing a number of weeks):

- Prompt the user for the required number of weeks to be entered, then complete the tasks described above for the required number of weeks *(in other words, get 5 valid readings for each week)*
- Include the sequence number of each week in the output (e.g. `week 1`).

```
How many weeks' temperatures needs to be processed? 2
Enter a temperature: 24
Enter a temperature: 18
Enter a temperature: 17
Enter a temperature: 16
Enter a temperature: 21

Week 1:
The average of the temperatures: 19.2
The number of cold days        : 2

Enter a temperature: 24
Enter a temperature: 23
Enter a temperature: 21
Enter a temperature: 22
Enter a temperature: 25

Week 2:
The average of the temperatures: 23
The number of cold days        : 0
```

Optional Task 2 *(*copy your code from task 1 and call it P7T2, *then edit your code)*

Adapt your program from task 1 to also do the following:

- Reads in a description for each week *(e.g. April Week 1)*,
- Calculates and displays the highest temperature for each week – you MAY NOT use of any build-in Math methods in your solution,
- Finds and displays the highest overall temperature, in your display you should also display the description for the week when the highest temperature was measured.

You may consider creating additional user defined methods in order to complete this task.

Prac marking rubric (prac 7)

Your prac will be assessed by the assistants on duty (in the actual practical session), based on the following rubric:

<table>
<tr><th></th><th>Mark</th><th>Description of mark allocation</th></tr>
<tr><td rowspan="4">Task 1 / 10
Create a user defined method called <code>GetTemp()</code> that takes no input (as an argument) and returns a valid temperature value between <code>-10</code> and <code>45</code> (a temperature value can have decimal places). If the user enters an invalid value, the method should repeatedly prompt the user for a value until a valid value is entered. When the method has a valid value, it should return that value to the calling program.
Write a user defined method named <code>IsCold</code> that can be used to determine whether a temperature can be classified as cold (less than <code>18</code>). The method must take as input a temperature, and return a <code>boolean</code>, indicating whether the temperature is classified as cold.
Write a program that reads in <code>5</code> temperatures by using the <code>GetTemp</code> method to ensure that valid temperatures are processed. You need to calculate the average of the temperatures. Your program must also use the <code>IsCold</code> method to determine the number of cold temperatures entered. When the <code>5</code> temperatures have been processed you need to display the average and number of cold temperatures (from the temperatures entered).
Now update your program to calculate the average temperature for each week (comparing a number of weeks):
• Prompt the user for the required number of weeks to be entered, then complete the tasks described above for the required number of weeks (in other words, get 5 valid readings for each week)
• Include the sequence number of each week in the output (e.g. <code>week 1</code>).</td><td>0</td><td>Not implemented, or does not compile</td></tr>
<tr><td>1</td><td>No method created, all code in main</td></tr>
<tr><td>3</td><td>Methods created, but not according to instructions</td></tr>
<tr><td>6</td><td>Methods created and correct according to instructions.
Output correct for 1 set of readings (1 week)</td></tr>
<tr><td>10</td><td>Methods created and correct according to instructions.
Output correct for any number of weeks</td></tr>
<tr><td colspan="3">Optional Task 2 (no marks)
Adapt your program from task 1 to also do the following:
• Reads in a description for each week (e.g. April Week 1),
• Calculates and displays the highest temperature for each week – you MAY NOT use of any build-in Math methods in your solution,
• Finds and displays the highest overall temperature, in your display you should also display the description for the week when the highest temperature was measured.</td></tr>
</table>