**Student Written Assessment (Assignment)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Business Unit/Work Group** | IT Studies | | |
| **Qualification Code** | National Code: ICA30115  National Code: ICT40115  National Code: ICT40815 | **Qualification Title** | Certificate III in Information, Digital Media and Technology  Certificate IV in Information Technology  Certificate IV in Digital Media Technologies |
| **Unit Code/s** | ICTPRG301 | **Unit Title/s** | Apply introductory programming techniques |
| **Assessment Task Title** | ParkingFee Assignment | | |
| **Student Name** | Submit your solution via your LEARN account | **Student SIS ID** | Submit your solution via your LEARN account |
| **Assessor Name** | You have been added to a LEARN group which defines your assessor. This is normally your Course Registration Number (CRN) lecturer. | **Date** | 2018 Semester 1 |

|  |  |
| --- | --- |
| **Student Guide for Written Assessment (Assignment)** | |
| **Overview of Assessment** | This assessment will require you to complete 3 versions of the ParkingFee assignment.  In this assessment you will cover the following topics:   * Use of sequence structures * Use of selection structures * Use of iteration structures * Use of arrays * Reading and writing of Text files   The assessment is broken up and assessed in 3 parts and you will submit each part gradually as you develop your skills through the course. |
| **Task/s to be assessed** | This is a practical task and will require you to code, test and document in the C#.NET language. The following pages will define the specific tasks you need to complete. |
| **Time allowed** | You have the whole course duration to complete this assignment, however each part has its own submission date.  The submission dates for each part are provided in the LEARN course and summarized in the Assessment Submissions topic in LEARN. |
| **Location** | You can complete this assessment during your practical sessions and at home. |
| **Decision making rules** | To receive a satisfactory outcome for this assessment you must complete all parts correctly. |
| **Assessment conditions** | You must follow the provided [ITWorks C#.NET coding standards](https://learn.tafesa.edu.au/mod/resource/view.php?id=446265) in completing this assessment. Any clarification for this assessment can be obtained through class time, practical sessions or online connect sessions with a lecturer. |
| **Resources required** | To complete this assignment, you will need to use Visual Studio 2015. Visual Studio 2015 and Windows based machines are provided in your practical classes. You can use a Mac if you prefer but these are not provided in the classrooms. |
| **Results/Re-assessment** | You will be provided feedback for each part of the assessment and be given the opportunity to resubmit with any required corrections only once. |

**Parking Fee Assignment**

You have been employed by the ITWorks organisation as a junior C#.NET programmer. Your job is to create some programs for a client that owns a car park to calculate the parking fee. The program requirements are given below.

**Assignment Part 1 – ParkingFee1**

## Scenario

The ITWorks client has asked for a program to be developed that meets the following requirements:

## Program Specification – Parking Fee 1

**Description**

The parking fee in a parking station is calculated on the whole number of hours multiplied by the hourly rate of $2.50. A program is to input the whole number of hours from the keyboard, calculate and output the parking fee to the screen. There is a maximum parking fee of $20.00.

**Inputs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Data Type** | **Source** |
| hours | Hours parked | integer | Keyboard |

**Processing**

Calculate and output the parkingFee given an hourly rate of $2.50, the maximum parkingFee is $20.

**Outputs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Data Type** | **Destination** |
| parkingFee | Cost of parking | decimal | Screen |

## Assessment Requirements

*Complete the following Assessment Task by modifying/re-writing the C# code from the ParkingFee quiz to cover the following Program Specification:*

## Design:

You need to provide a solution to the problem using pseudocode to document your design. Add the pseudocode as a comment at the top of your code.

## Coding:

Write the C#.NET code for the given Program Specification.

Note: Use ‘named constants’ for fixed amounts e.g.

const decimal HOURLY\_RATE = 2.50m;

const decimal MAX\_FEE = 20.00m;

Include at the top of your output screen:

ParkingFee1 program developed by: “add your name here”

## Code Documentation:

Include as a comment at the top of your code

your name

date

assignment parkingFee1

and the pseudocode.

Within your code include a comment to explain how the parking fee is determined, include in your explanation the purpose of using MAX\_FEE.

## Best Practices:

Follow the provided ITWorks organisational guidelines for developing maintainable code, and adhere to the provided ITWorks C#.NET coding standards, all documented in the ITWorks C#.NET Coding Standards document.

## Testing:

**Sample Test Data:**

An input of 7 hours will give a parking fee of $17.50, and input of 10 hours will give a parking fee of $20.00 (i.e. the maximum fee).

Once you have written your code and checked it works correctly with the above sample test data, change the MAX\_FEE=10 and see if you get correct results (7 hours should now give a parking fee of $10). You should not have to change any other code in your program other than the MAX\_FEE.

Create a word document containing screen shots of the running of your program with the test data suggested above with MAX\_FEE = 20.   
Name your word document ParkingFee1\_Testing\_YourName.docx.

## Client Sign-off:

Obtain approval from the client that your final code meets the above specifications. Your lecturer will act as the client and this sign-off will be provided in the marking guide on learn.

## Submission summary:

* Zip file of your C# Console Application (do not just submit the Program.cs file) called ParkingFee1\_Program\_YourName.zip.
* Word document called ParkingFee1\_Testing\_YourName.docx containing screen shots of the running of your program.

## Assignment Part 2 – ParkingFee2

## Scenario

The ITWorks client has asked for an updated version of the program to be developed that now meets the following requirements:

## Program Specification

**Description**

The parking fee in a parking station is calculated on whole number of hours multiplied by the hourly rate of $2.50. A program is required to input and validate the hours to be between 1-24 inclusive. Calculate the parking fee given there is a maximum parking fee of $20.00. Output the hours parked and the parking fee (formatted to 2 decimal places).

**Inputs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Data Type** | **Source** |
| hours | Hours parked | integer | Keyboard |

**Processing**

Input and validate the hours to be between 1-24 inclusive.   
Calculate the parkingFee given an hourly rate of $2.50, the maximum parkingFee is $20.  
Output the hours and parkingFee.

**Outputs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Data Type** | **Destination** |
| hours | Hours parked | integer | Screen |
| parkingFee | Cost of parking | decimal | Screen |

## Assessment Requirements

*Complete the following Assessment Tasks by modifying/re-writing the C# code from the ParkingFee1 code*

## Design:

You need to provide a solution to the problem using pseudocode to document your design. Add the pseudocode as a comment at the top of your code.

## Coding:

Write the C#.NET code for the given Program Specification above.

Note: Use ‘named constants’ for fixed amounts e.g.

const decimal HOURLY\_RATE = 2.50m;

const decimal MAX\_FEE = 20.00m;

Include at the top of your output screen:

ParkingFee2 program developed by: “add your name here”

## Code Documentation

Include as a comment at the top of your code

your name

date

assignment parkingFee1

and the pseudocode.

Within your code include a comment to explain the validation of the hours.

## Best Practices:

Follow the provided ITWorks organisational guidelines for developing maintainable code, and adhere to the provided ITWorks C#.NET coding standards, all documented in the ITWorks C#.NET Coding Standards document.

## Testing:

**Sample Test Data:**

Input 0 hours requires the hours to be re-input, 25 hours requires the hours to be re-input,   
5 hours would give a parking fee of $12.50. 10 hours would give a parking fee of $20.00.

Create a word document containing screen shots of the running of your program with the test data suggested above.   
Name your word document ParkingFee2\_Testing\_YourName.docx.

## Client Sign-off:

Obtain approval from the client that your final code meets the above specifications. Your lecturer will act as the client and this sign-off will be provided in the marking guide on learn.

**Submission summary:**

* Zip file of your C# Console Application (do not just submit the Program.cs file) called ParkingFee2\_Program\_YourName.zip.
* Word document called ParkingFee2\_Testing\_YourName.docx containing screen shots of the running of your program.

## Assignment Part 3 – ParkingFee3

## Scenario

The ITWorks client has asked for another parking fee program to be developed that meets the following requirements:

## Program Specification

**Description**

A computer program is required to read the daily hours parked by customers and output the hours and parkingFee for each customer to a file. There are 50 entries for customer’s parking hours (integer numbers) stored in a text file ‘hours.txt’. You are required to read the file and store the data in an array, there is no need to validate the data, assume all hours have already been validated between 1-24. Calculate the parking fee for each customer, there is a maximum parking fee of $20.00. Output the hours and parkingFee (to 2 decimal places) to file. Calculate and output the average (to 2 decimal places) of the parking fees to the screen.

**Inputs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Data Type** | **Source** |
| hours | Hours parked | integer | File |

**Processing**

Read the hours from file into an array of hours. Calculate the parkingFee given an hourly rate of $2.50, the maximum parkingFee is $20.

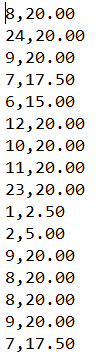
Output the hours and parkingFee to file with a comma separating the values.

Output the averageParkingFee to the screen.

**Outputs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Data Type** | **Destination** |
| hours | Hours parked | integer | File |
| parkingFee | Cost of parking | decimal | File |
| averageParkingFee | The average of all parking fees | decimal | Screen |

Output the hours parked and the parking fee (to 2 decimal places but no $ sign) to a text file called parkingFee3.txt. Include a comma between the values. A sample of the file is as follows:



## Assessment Requirements

## Coding:

Write the C#.NET code for the given Program Specification above.

Note: Use ‘named constants’ for fixed amounts e.g.

const decimal HOURLY\_RATE = 2.50m;

const decimal MAX\_FEE = 20.00m;

Include at the top of your output screen:

ParkingFee3 program developed by: “add your name here” and output the average parking fee.

## Code Documentation

Include as a comment at the top of your code

your name

date

assignment parkingFee3   
(you do not need to include your pseudocode solution for this assignment).

Within your code include a comment to explain the data is read from file and stored into the array.

## Best Practices:

Follow the provided ITWorks organisational guidelines for developing maintainable code, and adhere to the provided ITWorks C#.NET coding standards, all documented in the ITWorks C#.NET Coding Standards document.

## Testing:

Create a word document containing a screen shot of the running of your program to display the averageParkingFee. Name your word document ParkingFee3\_Testing\_YourName.docx.

## Debugging:

Include in your word document two screen shots demonstrating that you can correct errors by stepping & tracing through your code.   
Insert a break point at the Console.ReadKey() at the end of your code. Add one screen shot showing the watch window at the beginning of stepping through your code. Continue to the break point and add another screen shot of the final watch window to verify your averageParkingFee result is correct.

## Client Sign-off:

Obtain approval from the client that your final code meets the above specifications. Your lecturer will act as the client and this sign-off will be provided in the marking guide on learn.

**Submission summary:**

* Zip file of your C# Console Application called ParkingFee3\_Program\_YourName.zip.
* Word document called ParkingFee3\_Testing\_YourName.docx containing a screen shot of the running of your program to display the average parking fee.