T: Sean Shearing

Procedural Programming

Ashley.Colman

Contents

[Introduction 2](#_Toc400463811)

[Key heading one 2](#_Toc400463812)

[Key heading two 2](#_Toc400463813)

[Key heading three 2](#_Toc400463814)

[Conclusion 2](#_Toc400463815)

[References 3](#_Toc400463816)

Unit 16: Procedural Programming

# Introduction

In this report I will go into detail about the program I have developed during this unit, explaining the good points, bad points, whether or not it meets the requirements set out to achieve, how it meets these requirements and how I could possibly improve the program if given more time.

## Good Points

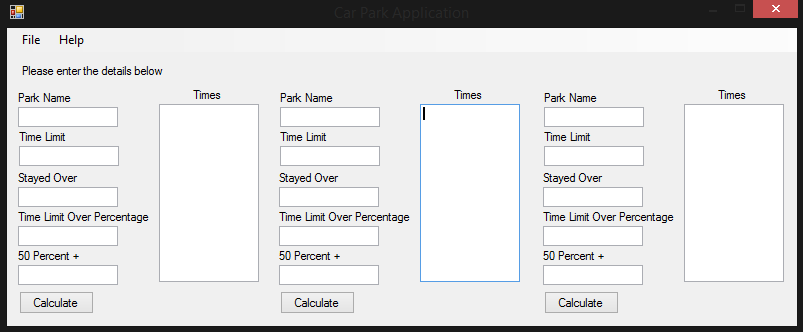
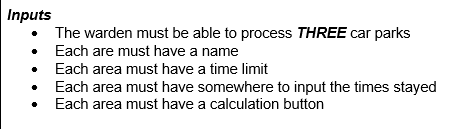
The first good point of my application is that when the run button has been clicked the program runs without any problems, no freezing, bluescreening or providing an error message to the user. This is good as it means that the user will be able to run the application and use it as intended without having to try multiple times in order to get it to work, saving time and leaving the developer a good reputation. When the user runs the program they will be brought to the car park program screen that will display 3 car parks as intended, including the required information for working out how many cars stayed over the time limit and whether or not a warden needs to be sent out. Each of the textboxes that require the user to input data into (txtCarParkName, txtTimes and txtTimeLimit) allow the user to input any text they like as the textbox allows freehand text. Because of this the user can use the application to input different names, times and time limits making the application useful for many different scenarios and can be used with many different inputs depending on what the user enters. When the calculation button is clicked the information that has been entered by the user is processed and outputted to the correct textboxes (txtAllotted, txtOverPercent and txtWarden. The information that has been process is also correct meaning that the program works as intended and gives reliable information which is what the user wants from using the program. The program also uses functions to convert data, work out how many cars stayed over the time limit, work out the percentage of cars that stayed over and whether or not a warden needs to be sent out based on the percentage. This is good as it means that the code has been simplified and made more efficient. The same code is reused for each of the car park programs making it easier for the developers to know what has been written and less code has been written overall. This can also make the testing stage of the program much easier as there is less code to check and less problems are likely to occur. Another good point to the program is that all three of the car parks work. This is good as it means it has met the specification that I have tried to meet and the user can use the application to calculate data for three individual car parks on one program. The ability to input data into three different car parks on one program means that the user won’t have to open the application, enter details, calculate, close the application and then open it back up in order to work out multiple car parks. Instead the user can input the data into all three of them and click calculate, saving time for the user. Another good point the program has is that the menu strip tabs work correct, displayed a dropdown list when clicked and have multiple buttons inside them. The reason for why this is good is because it means that the user can get access to even more tools from simply clicking on the tab without having to make the program more cluttered and taking up space by adding more buttons to each of the three car parks. When the file tab is clicked a dropdown list is opened up containing three save buttons, three load buttons and exit. The inclusion of these buttons is good as it means that user can save the information inputted/outputted in each individual car park program and then load the information back when needed. Having a save and load options was also one of the requirements the program had to meet in order to pass the assignment too. The exit option means that the user can close the application at any time from simply clicking the button without any problems occurring. The menu strip also has a Help tab that when clicked opens up a dropdown list containing a launch button. The inclusion of the launch button is good as it provides an online help webpage containing information on how to use the program once clicked. The webpage also loads without any problems and was one of the requirements the program had to meet. The program also includes tooltips on every control within the program such as textboxes, buttons and tabs. This is good as it means that the user will understand what each control does/is used for on the program to further assist them if they’re confused with how it works. It can also help people that aren’t very experienced with computers use the program or people who aren’t able to access the help webpage due to not having an internet connection. Another point is that the labels used on the program lets the user know what each textbox is used for/displays.

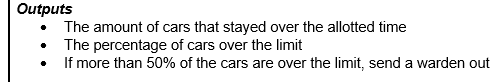
## Bad Points

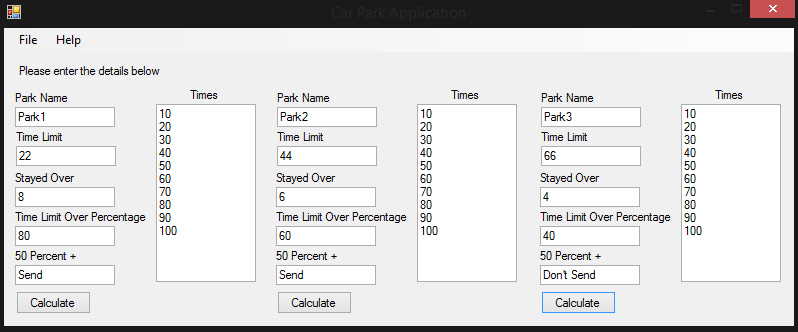
The first bad point of the program is that there is no way the user can access the help document if they’re unable to connect to the internet, unless they have the document with the program and know where to find it in the directory. This is bad as it means people who don’t know how to use the program will either struggle or simply won’t be able to use it. The tooltips can help the user however the help webpage has a more in-depth guide on how to use it and people who are not experienced with computers might not be able to understand how to use the program based on the tooltips alone. The second bad point to the program is that there is no support to aid users with visual/hearing impairment interacting with the program. The program doesn’t include a magnifier/audio reader tool for increasing the text size/reading the text within the program to the user. The reason for why this is bad is because it means that people with disabilities won’t be able to use the program at all, less money will be made if the program was to be sold to the public and the company/developers reputation could be affected by not including it. The third bad point of the program is that when the user has loaded the file information back into the program it is only outputted into the txtTimes textbox. The reason for why this is bad is because although the information is outputted into 1 textbox all of the information is outputted into it, instead of outputting each of the information into each correct textbox. For example outputting the time limit into txtTimeLimit and the Percent into txtOverPercent instead of outputting all of it into txtTimes. By outputting all of the information into txtTimes it requires the user to then have to input the information into the correct textboxes again once loaded, instead of the load doing it for them automatically. The final bad point to my program is that the program hasn’t used 1 function for saving/loading and has instead used multiple for each of the three car parks. The reason for why this is bad is because more code has been written by the developer making it more complex and less efficient than it needs to be.

## Does it meet the requirements you set out to achieve and how?

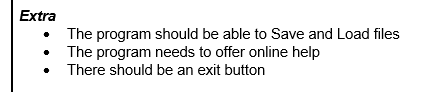
The program that I have developed for the past few weeks does meet the requirements that I set out to achieve. The way in which my program meets these requirements is by having a fully working error free application that allows the warden to process three car parks. When the application is opened three car parks are displayed to the user containing labels, textboxes and buttons allowing the user to input the car park name, time limit, time stayed and then calculate how many cars stayed over, percentage and whether or not a warden needs to be sent out for each individual car park.

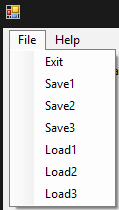
The program meets the input requirements specified in assignment 3 by allowing the user to input a name into txtCarParkName (txtCarParkName2, txtCarParkName3), time limit into txtTimeLimit (txtTimeLimit2, txtTimeLimit3) and times stayed into txtTimes (txtTimes2, txtTimes3). Once the information has been inputted the user can then click the calculation button (btnCalculate, btn Calculate2, btnCalculate3) to then calculate the information inputted. The user can also input the information into the other two car parks if they would like to and then click the calculate button, the only difference is the textbox the information will be inputted into. 

The program meets the output requirements specified in assignment 3 by including textboxes that output the numbers of cars that stayed over the allotted time, percentage of cars over the limit and whether or not a warden needs to be sent out. For the cars that stayed over the information is outputted in txtAllotted (txtAllotted2, txtAllotted3), for percentage the information is outputted into txtOverPercent (txtOverPercent2, txtOverPercent3) and for whether or not a warden needs to be sent out the information is outputted into txtWarden (txtWarden2, txtWarden3). Based on what car park the user has inputted the information and clicked calculate will change where the txt outputs will be. The program also meets the outputs requirements as the information that is outputted is correct based on the information inputted in the input textboxes. 



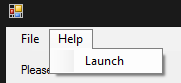
The program meets the extra requirements specified in assignment 3 by including save, load, help and exit options for the user. The save, load and exit options are located within the File tab of the menu strip, when clicked the user can choose to save the information from the first car park, second car park or third Cark Park or load previously saved information back into each car park. When the car park has been calculated the user can save the information to a txt file called data1.txt (data2.txt, data3.txt). If the user would like to close the application the user can click on the exit button from within the File tab and it will close the program without any problems. There is a Help tab within the menu strip that when clicked displays a launch button, when clicked an online help webpage is opened up in the users default browser where they can read how to use the program.



Save options

Load options

Exit option

Help option

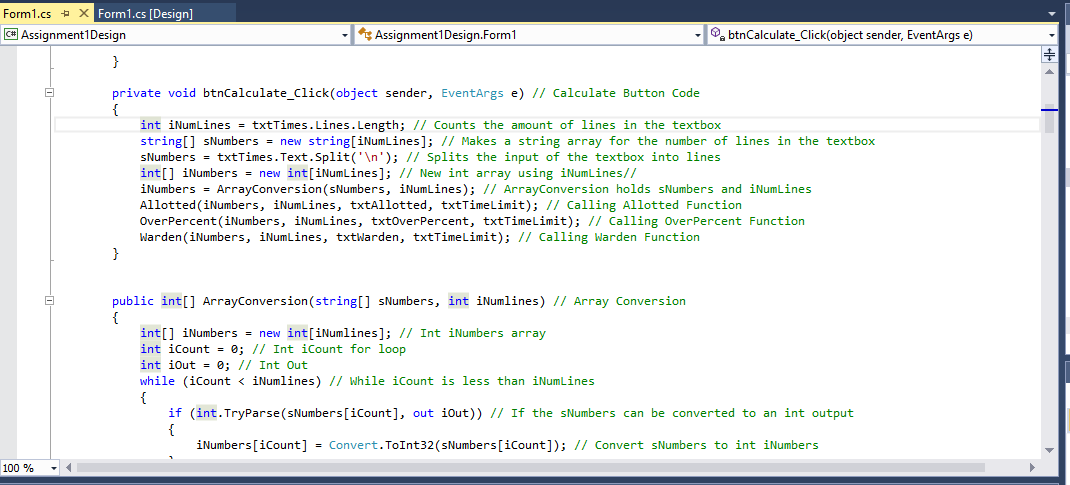
## https://i.gyazo.com/fd96f97452312042097514e9e411a6f7.png

Online help webpage screen

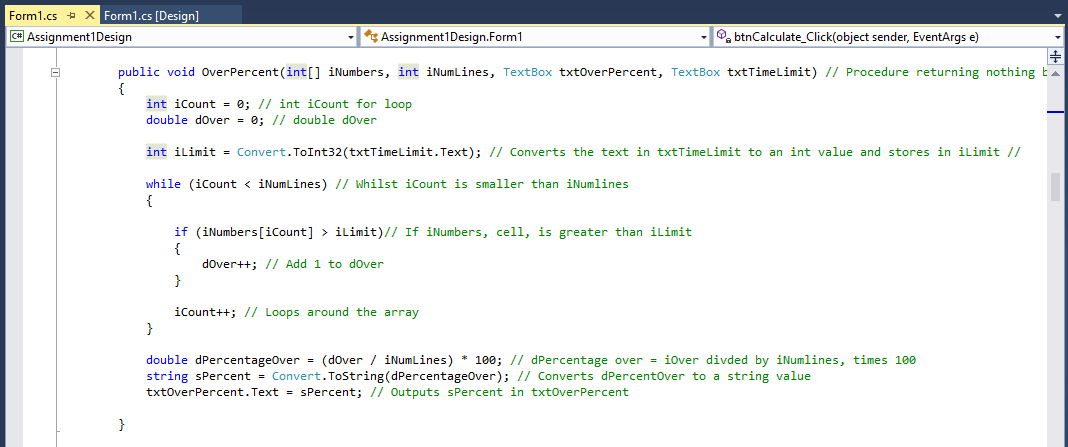
The program also includes tooltips for every control on the program. When the user hovers over a textbox, button or tab a tooltip will be displayed to the user informing them what the control is used for, what needs to be inputted or what is outputted there. One of the requirements was too include tooltips in the program.

https://i.gyazo.com/ff8322ca4c395b8a05d5092ffabf4fd2.png

Another way in which my program has met the requirements is by ensuring that all of the code written within the program has been commented. Since the code for my program including the functions, procedures and variables it means that the program can be developed on in the future as stated in assignment 3. The reason for why this can allow the program to be further developed is because the developers who are going to work on it in the future will be able to understand the code I have already written due to the inclusion of my comments.



My program has also met the requirement of being a procedural program through using procedural methods. In order to make the program more efficient by reducing the amount of code written by the developer and used within the program I have used procedural programming methods, functions and procedures. In order to calculate how many cars stayed over the time limit, percentage and whether or not a warden needs to be sent out I used a function which is then used by all three car parks. The function is then called within the button for each car park and calculates the information for each individual car park from simply changing the textboxes within. The program has met being a procedural program from using this.



## How could you improve the program?

The first way in which my program could be improved more effientmply changing the textboxes within. ogram can be s outputted the