

Soft152 Assignment– Academic Year 2019/2020

1 Assignment Information

1.1 General Information

- Module Leader: Dr Liz Stuart.
- This assignment is worth 100% of the marks for this module as specified in all documentation. Note there are four stages to handing in this work! See section 1.2 for full details.
- Please pay attention to the following notes on coursework submission.
- Note that work submitted up to 24 hours after the deadline will be capped at 40%.
- Work submitted more than 24 hours after the deadline will receive an automatic mark of zero.
- If you have valid extenuating circumstances for late work, the policy has not changed - the actual mark achieved will be recorded against that submission, no capping will apply”.
- *The Faculty staff have asked me to make you aware of the following: Please be advised that I plan to mark and return your work quite quickly (for some it will be within a few days of you submitting it). Due to this tight schedule, students who are deemed to have validated extenuating circumstances will **not** get the usual extension of 10 working days. Such students will only be entitled to an extension until Monday 13th January 2020 (the date the vivas begin). Please contact the Faculty for more information.*

1.2 Module Learning Outcomes being assessed

- Demonstrate an understanding of software engineering.
- Apply good programming practice by using a structured, methodical approach to software development.
- Design and implement programs of moderate size and complexity.
- Plan and implement testing and verification of software.

1.3 Assessed Skills Elements

- Knowledge and Understanding (LO2, LO3, LO4)
- Synthesis/Creativity (LO2, LO3)
- Evaluation (LO3, LO4)
- Problem Solving (LO2, LO3, LO4)

1.4 Important Dates

- Data file released: Thursday 14th November 2019 (week 16). Note that the data file is not required for stage 1.
- Stage 1 of the submission - This is assessed by Liz during your lab in week 17
- Stage 2 of the submission - This is assessed by Liz during your lab in week 20
- Stage 3 of the submission – DLE eSubmission due on Wednesday 8th January 2020 at 12:00
- Stage 4 of the submission - During the examination and assessment weeks 25 and 26, you will be required to attend a viva.
- Marking Deadline: Wednesday 5th February 2020.

2 Assignment Specification

2.1 University Elements and Components

The assessment for this module is made up of two elements: a C1 coursework element and a P1 practice element.

The C1 coursework element is weighted at 35% for this application.

- It is made up of two components.
 - The first component is called “Classes coded” and it is weighted 15%.
 - The second component is called “Data read into array” and it is weighted at 85%.

The P1 practice element has one single component (a viva) weighted at 65%.

2.2 Shall we talk ‘Marks’ instead?

To help us all understand all this terminology, I will rephrase section 2.1 in terms of marks, as I hope it will simplify all the jargon. The max number of marks for the application is 100 marks.

- C1
 - Classes written – worth 5.25 marks
 - Data read in – worth 29.75 marks
- P1
 - eSubmission + Viva – worth 65 marks

2.3 Understanding what you have to do and when

Fundamentally, this means that the assessment of the entire module is based on one assignment. However, you will be required to submit the code for the assignment in four stages. The reason for doing this is to ensure you complete your code in a timely professional manner.

2.3.1 Stage 1

- Stage 1 is more like a check. You are required to show your code for the three main classes during your week 17 lab.
- The time will depend on which session you are allocated to. If you are unsure, please ask.
- Each of your classes should comprise of all attributes of the class, at least one constructor as well as all the get and set methods for the class attributes (not in shorthand).
- You must attend your own lab session that week to be assessed!

2.3.2 Stage 2

- Stage 2 is quite a substantial part of the coursework. You are required to show me your assignment running during your scheduled C# lab with Liz Stuart in Week 20.
- The time will depend on which session you are allocated to. If you are unsure, please ask.
- By that stage you are required to have set up an appropriate array data structure to store your data and to have read the input file into the array, as specified. I will dynamically inspect the contents of your array. You should not use <Lists> or any other collection supplied by C#.
- You must attend your own lab session that week to be assessed!

2.3.3 Stage 3

- Stage 3 is the **electronic submission** of your assignment on the DLE before or by the deadline in week 24. The deadline is **Wednesday 8th January 2020 at exactly 12:00**.
 - This is the submission of your entire coded C# application. You must take the entire application folder and **zip** it up into an archive file (no .rar files will be accepted).
 - If you fail to submit your code, I cannot mark your code quality and you will lose all the corresponding marks in the assignment’s marking scheme.

2.3.4 Stage 4

- Stage 4 is a formal demonstration and viva voce (oral examination where you will be asked questions on your code/design choices).
 - Note that this event is weighted at 65%.
 - This will take place during weeks 24 and 25. This is only a few days after you submit your work; so be ready! It will last around 20 minutes.
 - You MUST bring your code and your whole application with you to demonstrate.
 - If you fail to arrive at your viva in plenty of time to login, load and then complete your viva, **you will be awarded a mark of zero** for the P1 element.
 - Please be careful to arrive early enough to give yourselves time to login and set up your application. I advise arriving no less than 20 minutes early to give you time to login, load your code and test it all before we start.
 - I also very strongly recommend that you test your code works correctly in the lab (tbc) that we will be using for the vivas well before the day of your viva.
 - Many students find their code does not run correctly minutes before their viva begins. Please try not to let this happen to you!
 - Note the lab I am allocated will not be available during the viva to other students, so you need to get organised!

2.4 More on the Viva

The viva is a compulsory event and it is mandatory. To reiterate, **the entire P1 practice element will be awarded a mark of zero if you fail to attend your specified viva slot. No exceptions.** This is the sole event at which you will demonstrate the functionality of your system; so be well prepared! I recommend that you practice by running your software on, at least two, standard Plymouth University machines.

During the viva, you will be asked to demonstrate some/all of the product functionality. I may also ask you to show me portions of your code, the contents of your input text file and other aspects of your product. Note that if you do not show me something working during your viva, it will receive no functionality marks. I will not be marking the electronic copy of your work for functionality, (I will use it solely for marking code quality) – this is your one and only opportunity to get functionality marks.

Each viva will be allocated (approximately) 15 - 20 minute slots during Week 25 and 26.

- I will publish the viva schedule nearer the time on the DLE.
- You must be available from 9am-6pm each day (I will work around any examinations you have).
- Earlier and extra slots
 - If you are interested in an early am slot between 8-9am please mail Liz!
 - I will probably also be available to do vivas in week 24 the day after you submit.
 - If interested please email Liz before stage 4 submission; these are subject to availability (first come; first serve basis).

2.4.1 Can you help me out?

Time will be limited, so **please be prepared to be brief** and demo all these actions quickly and efficiently! Please do not hold me up with long discussions and explanations of any problems you have had. Come and see me in my office at another time if you need to talk to me about any other matters. I will not have time during the viva schedule and I do not want to come across as being rude but I will stick to my schedule. **I appreciate your understanding in this matter!**

3 The assignment scenario

3.1 The Data

Since 2008, guests and hosts have used Airbnb to expand on traveling possibilities and present more unique, personalized way of experiencing the world. This dataset describes the listing

activity and metrics in NYC, NY for 2019. This dataset has all the data you need to find out more about hosts, geographical availability and other necessary metrics to make predictions and draw conclusions. This data was sourced from <https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data/downloads/new-york-city-airbnb-open-data.zip/3>. Note it has been formatted specifically for your coursework so do not use the original source. Instead, you must only use the one I provide via the DLE.

The application is intended for use by members of the public who are interested in analysing Airbnb data as well as Airbnb staff who are responsible for making sure data is accurate. The assignment will focus on data collected for New York City.

All admin staff at the Airbnb office will use the system in the same way. They will need to be able to do the following tasks:

DISTRICT

- SEARCH and DISPLAY district data appropriately
- ADD a new district
- EDIT a district name

NEIGHBOURHOOD

- SELECT and DISPLAY neighbourhood data appropriately
- ADD a new neighbourhood
- EDIT a neighbourhood name

PROPERTY

- SELECT and DISPLAY a property
- ADD a new property
- EDIT/DELETE an existing property

A **District** is defined by the following characteristics or features:

- District name
- The number of neighbourhoods in that district
- All the neighbourhoods within that district (an array of **Neighbourhoods**)

A **Neighbourhood** object is defined by the following features/characteristics:

- Neighbourhood name
- The number of properties in that Neighbourhood
- All the properties within that neighbourhood (an array of **Properties**)

An **Property** is defined by the following features/characteristics:

- Property ID
- Property Name
- Host ID
- Host Name
- Number of properties listed for this host
- Latitude
- Longitude
- Room Type
- Price
- Minimum number of nights
- Availability number of days; out of 365 days per year)

3.2 Adding functionality to support analysis of the data

Once you have completed the main functionality for the application (defined above) you should move on to the final 'Analysis tools' part of the assignment.

The functionality of this part of the assignment is entirely up to you. This is your opportunity to design, create and code appropriate, relevant and helpful functionality. This is an area where you can use your programming flair to devise unique and innovative data displays.

3.3 Input data file

An input data file for the system will be provided for you in due course. It will be available from the Soft152 DLE. This file contains all the information you will need to implement the system.

Your system must read all of the input data from the input file and store it appropriately using ONE SINGLE array data structure for data storage. Note there SHOULD be other arrays embedded inside this main array. You should not have several individual arrays. No other method of internal storage of this data is permitted!

Please talk to me about this if you have any concerns whatsoever.

Once the input data has been read into the system, the data should be made available to the user in an appropriate manner. The design of the form(s) and how you support the functionality required is an area you can focus your design skills. Remember: You must try to think as the user will think and try not to make assumptions if possible but stay within the brief at all times. You should confine your skills and expertise to producing the best system as specified.

3.4 Marking scheme

This assignment will be assessed out of 100 marks.

Assuming that your system reads in the data from the file correctly this will be divided up into

- Coded classes – 5.25 marks
- Data read into array – 29.75 marks
- Viva – 65 marks
 - **Functionality** of the product (subset marked during your viva)
 - District actions – Search and Display, Add and Edit
 - Neighbourhood actions – Search and Display, Add and Edit
 - Property actions – Search and Display, Add, Edit and Delete
 - Added analysis and graphics functionality (show off your flair here!)
 - **Quality** of the implementation – marked using your hardcopy submission of your C# code
 - Implementation/Coding style
 - Interface, Usability and User feedback
- If your system does not read in the data from the file an alternate marking scheme will be applied. Note that quality marks are weighted by the amount of working functionality.

4 You must not take 'shortcuts'

- The data must be stored in a single array.
 - You may **not** use the <LIST> or other collections supported by C# (sorry).
 - If using arrays, then only the .Length property may be used.
- The application should not make use of C# build in functions where their use is solely to provide you with a shortcut; to enable you to avoid coding the functionality yourself.
 - **get; set;** shortcuts in class declarations (If you don't know what this means, you are already not using this shortcut!)
 - the **.Contains** in the search functionality to you code
- If you are coding the EXTENDED application then do not use any C# chart components for drawing graphs.
 - For the bars use the standard graphics methods of DrawRectangle() or FillRectangle().
<https://msdn.microsoft.com/en-us/library/system.drawing.graphics%28v=vs.110%29.aspx>
 - Please do not hard code an absolute path to the folder which contains the data files.

- An example of an absolute path is something like:
“C:\Users\Documents\Assignment\heathrow.txt”.
- Use a file dialog component instead.

It is very difficult for me to provide a definitive list; so please if you are unsure; **ASK LIZ**, I am happy to advise on this.

5 Assessment Criteria

This assignment will be assessed on the following:

- Program **design** - program interface is easy to use and provides an appropriate quantity of user feedback!
- Program **execution** - program executes correctly without errors program runs in a logical manner
- Program **clarity** - suitable names for variables etc., appropriate use of comments*, general layout (indenting etc.), suitable names for forms and components
- C# facilities - appropriate use of interface components, appropriate use of language overall
- You **must not change the structure** of the input data file (of course you can change the data by editing data and adding data as specified in the functionality lists). However, please bring the original data file with you to the viva. I reserve the right to ask you to use a different data file on the day of your viva. You have been warned.
- Additionally, **you must not add** to the functionality of this brief. However, do please demonstrate your programming flair by using
 - good **quality** code - comments, consistent indentation, well named variables
 - input **validation**
 - **modularity** - use shorter methods and pass parameters rather than great big long event procedures.
 - a **consistent interface** across the whole product – this product should not look like it was programmed by four different people
 - gathering/using **user feedback** to improve your product.

This is part of the module in which I really get the opportunity to examine how you are programming. I will be assessing how you have programmed your solution in detail - so please do pay attention to this. A program that just works will not achieve the maximum marks. The solution must also be programmed well, in a modular structure. That includes good naming of variables, standardised use of component prefixes (e.g. txtSurname is a good name for an input text box for a customer's surname).

5.1 Penalties that will be applied to your final result

It is also possible to lose marks from the total mark you gain for the assignment. These details have already been covered but are re-iterated here for clarity. You will lose marks if you

- Fail to present, on time, your coded classes – minus 5.25 marks
- Fail to present, on time, your code with data read into array – minus 29.75 marks
- Fail to upload code/attend scheduled viva - minus 65.

Having said all this, I do very genuinely want you to avoid these common pitfalls, so please pay attention to HOW and WHEN you submit your work. ASK, ASK, ASK if you are unsure ☺

6 Getting help

- I would recommend you adopt some version control whilst developing this solution. The most widely used application for this is GitHub.
- However, please note that this is not a collaborative piece of work. It is an **individual work assignment**, and must be your own work. Any plagiarism discovered will be dealt with using

the appropriate process – consult your student handbook if you are unaware of this as it could jeopardise your studies. It would be better not to submit at all rather than submit copied work. You have been warned!

- You should NOT START programming Stage 2 of your assignment until you have completed and understood both the BOOKS application and the LIBRARIES applications.
 - **This is the biggest mistake students make every year!**
 - I am in the process of adjusting the structure of the course to enable me to dedicate a full week to teaching/support to these most important applications of the module (as they will fully prepare you for the assignment)
 - I will continue with support on the BOOKs and the LIBRARIES in **Week 17 and 18**
 - I will now do assignment support lab sessions in **Week 19 and Week 20**.

I wish you all the very best.

Liz Stuart

Soft152 Module Leader