|  |
| --- |
| Bottom-Text co. |
| [Project C] |
| ICTPRG418 |
|  |
| **Callum Buchanan** |
| **4/3/2017** |

|  |
| --- |
| [This documentation contains a project plan, development methodologies and systems, coding standards and testing] |

Table of Contents

[Introduction 2](#_Toc479013579)

[Requirements 2](#_Toc479013580)

[Methodology 4](#_Toc479013581)

[Project Management Software 4](#_Toc479013582)

[Development Language 4](#_Toc479013583)

[Development Software 4](#_Toc479013584)

[Communication software 4](#_Toc479013585)

[Coding standards 4](#_Toc479013586)

[Project Plan 5](#_Toc479013587)

[Risks 5](#_Toc479013588)

[Assumptions, Constraints and Dependencies 5](#_Toc479013589)

[Project Management 5](#_Toc479013590)

[Communications Management Plan 6](#_Toc479013591)

[Project Team 7](#_Toc479013592)

[Staffing Management Plan 7](#_Toc479013593)

[Quality Management Plan 7](#_Toc479013594)

[Risk Management 7](#_Toc479013595)

[Future Requirements 7](#_Toc479013596)

[Testing 8](#_Toc479013597)

[Screenshots 9](#_Toc479013598)

[Automated Build Script 11](#_Toc479013599)

[Client Sign Off 11](#_Toc479013600)

Introduction

Bottom-Text co. has been asked to build a C++ application. The project being required to make has to:

-Create and manipulate a 2d array

-Sort 2d array

-Search array

-Create a “question” object

-Read text file which answers coding questions

-Binary file handling procedures

-Pass-by-referencing

-Uses random access algorithms

-Have a calculator built into the software

-Ability to change background color

-Ability to play music

# Requirements

Apply intermediate programming skills in another language

1 - Code using user-defined data structures

-1.1 Design, define and use data structures that are aggregate of other data types 1.2 Code using array of user-defined data types

-1.3 Use facilities in language to create, manipulate and destroy dynamic variables such as arrays

2 - Code using standard algorithms

-2.1 Use modular programming approach to coding, including pass-by-reference parameter passing

-2.2 Write code to create and manipulate 2-D array

-2.3 Create and maintain sorted array and use language-provided facilities for sorting array of ordered elements

-2.4 Code simple binary search technique for use with array of sorted data

-2.5 Code binary file-handling solutions using random-access algorithms

3 - Debug code

-3.1 Use stand-alone debugging tools or tools provided by integrated development environment (IDE) to debug code

-3.2 Use debugger to trace code execution and examine variable contents to detect and correct errors

4 - Document activities

-4.1 Follow organisational guidelines for developing maintainable code and adhere to provided coding standard when documenting activities

-4.2 Apply internal documentation, suitable for use by peers, to all code created.

-4.3 Use documentation tools available in target language when documenting activities

5 - Test code

-5.1 Design and document tests

-5.2 Undertake limited testing of produced code to ensure compliance with program specification

-5.3 Capture and record test results

6 - Create application

-6.1 Build application in response to user requirement

-6.2 Access multiple source-code files

-6.3 Employ IDE project maintenance facilities or make files to automate program building

-6.4 Develop program specification solution when provided with basic design document

-6.5 Design algorithm and document, construct and test applications in response to problem description using target language

-6.6 Document completed application

Manage a project using software management tools

1 - Select the software management tools

-1.1 Determine the software development methodology to be used

-1.2 Determine the project management software to be used

-1.3 Determine the source-control system to manage the source code and to handle conflicts

-1.4 Determine the collaboration software to be used

2 - Use the software management tools

-2.1 Create a project plan, according to the software requirement specifications

-2.2 Define the source-control procedures

-2.3 Create a collaboration environment

3 - Monitor the use of software management tools

-3.1 Monitor, and adjust, the project plan to maintain progress according to the project plan

-3.2 Ensure that the code is correctly entered into the source-control system

-3.3 Monitor the collaboration environment, and resolve issues where required

# Methodology

The methodology our team decided to work with was the Spiral development cycle.

## Project Management Software

-Trello

-gitHub

Source code is uploaded to gitHub repositories.

## Development Language

-C++

## Development Software

-Visual Studio 2015

## Communication software

-Slack

## Coding standards

-Name variables based on what their use is

-Camel case

-This style of indentation:

function()

{

//bits of code

}

# Project Plan

## Risks

None aside from Memory block errors that hamper development progress, but that can be easily dealt with

## Assumptions, Constraints and Dependencies

The Project C application only has 1 constraint in place:

* Software is a Windows 32bit C++ application

Assumptions are as follows:

* No UI has been asked so a console application is being developed.
* Question object does not require manipulation

Dependencies:

* Internet connection for team communication
* Visual Studio 2015

# Project Management

|  |  |
| --- | --- |
| Task Name | Project C |
| **Target Date** | 05/03/2017 |
| **Start Date** | 07/02/2017 |
| **Location** | North Metropolitan Tafe |
| **Milestones and Milestone Dates** | Using the SPIRAL software development framework the database will be created in this order:   1. Determine Objectives - 12/02/17 2. Identify and Resolve Risks - 17/02/17 3. Development and Test - 22/02/17 4. Review - 27/02/17 5. Implement the system– 05/03/17 |
| **Project manager** | Jack (don’t know his last name!) |
| **Company** | Bottom-Text co. |
| **Company Representative** | Jack – Project Lead |
| **Additional Staff** | Danny – GitHub admin  Jeff  Martin  Callum Buchanan |
| **Additional Staff tasks** | Slack communication |
| **Budget** | $100000000000 |
| **Funded By** | Centrelink |

## Communications Management Plan

This Communications Management Plan sets the communications framework for this project. It will serve as a guide for communications throughout the life of the project and will be updated as communication requirements change. Communication will commence through the company representatives via email, Slack online chat messenger and face to face meetings.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Communication Type** | **Description** | **Frequency** | **Format** | **Participants/ Distribution** | **Deliverable** | **Owner** |
| Weekly Status Report | Email summary of project status | Weekly | Email | Project Manager, Project Team | Status Report | Company Representative |
| Weekly Project Team Meeting | Meeting to review and assess | Weekly | In Person | Project Team | Minutes | Company Representative |
| Project Finalization | Finish project and release | As Needed | In person | Project Team, Server Maintenance | Final Package | Company Representative |
| Documentation Review | Review of any documentation | Monthly | Email | Project Manager, Company Representative | Documentation package | Project Manager |

## Project Team

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **E mail** |
| Jack | Project Lead | [JJ@bottom\_text.com](mailto:JJ@bottom_text.com) |
| Danny | GitHub Admin | [danny@bottom\_text.com](mailto:danny@bottom_text.com) |
| Callum Buchanan | Project Coder | [cb@bottom\_text.com](mailto:cb@bottom_text.com) |
| Martin | Project Coder | [JeffJohnson@bottom\_text.com](mailto:JeffJohnson@bottom_text.com) |
| Jeff | Server Maintenance | [JohnJoe@bottom\_text.com](mailto:JohnJoe@bottom_text.com) |

## Staffing Management Plan

Project Lead (1) – Responsilbe for ensuring programmers and gitHub admin are doing their work and meeting deadlines. Responsilbe for organising meetings

Github Admin (1) – Responsible for maintaining gitHub repositories

Programmers (3) – Responislbe for programming code and meeting deadlines

## Quality Management Plan

All member of the team will be responsible for the of quality management throughout development. It is of the upmost importance that the project is delivered to the quality standards of Bottom-Text co.

## Risk Management

No risks aside from memory block errors exist in the project as determined by it’s requirements. No management is required in this instance.

## Future Requirements

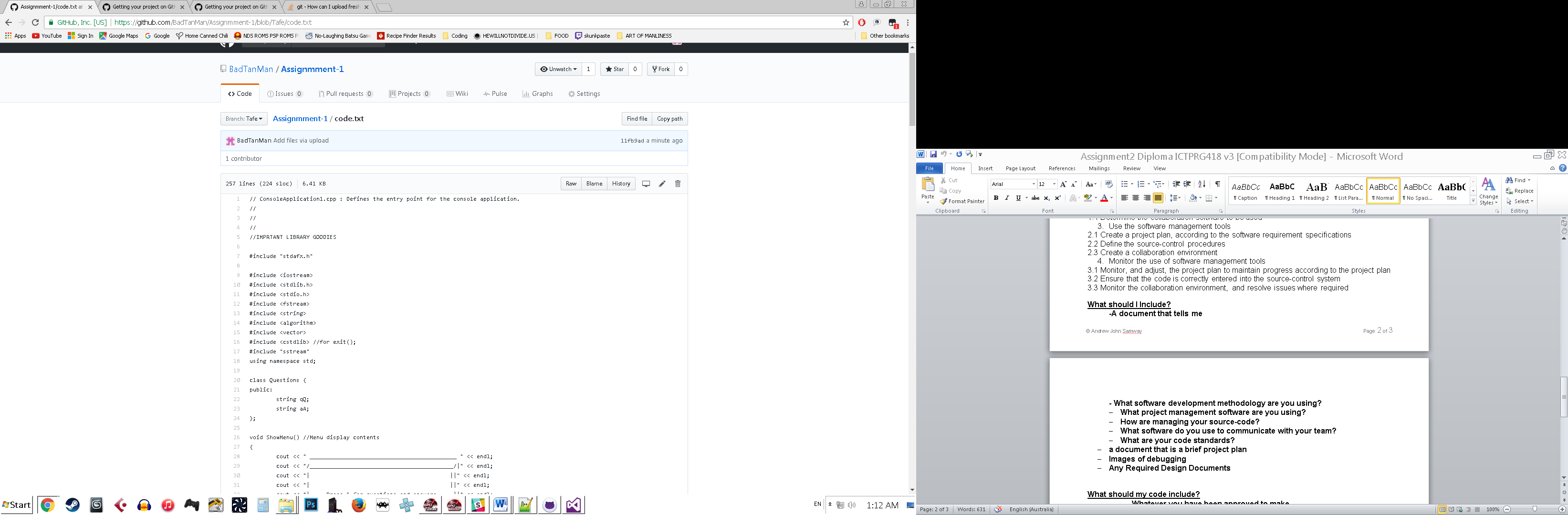
N/a.

# Testing

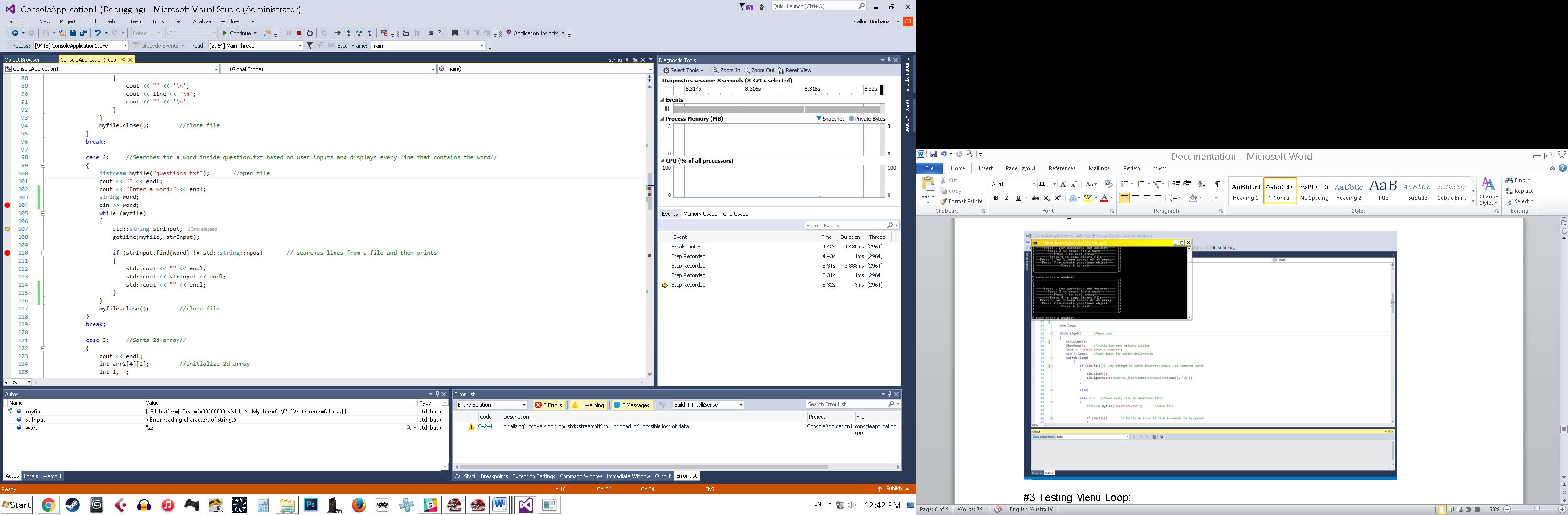
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No. | Description | Inputs | Data Type | Expected Output | Actual Output | Pass (Y/N)? |
| 1 | Menu input check | a | char | “Incorrect input!” | “Incorrect input!” | Y |
| 2 | Option 1 check | 1 | int | Output text file questions | Output text file questions | Y |
| 3 | Option 2  Search Word  Check | “the” | string | Outputs lines containing the word “the” | Outputs lines containing the word “the” | Y |
| 4 | Option 3  Check | 3 | int | Output 2d array and a sorted 2d array | Output 2d array and a sorted 2d array | Y |
| 5 | Option 4  Check | 4 | int | “bin file has been copied” | “bin file has been copied” | Y |
| 6 | Option 5  Check | 5 | int | “Enter a number to search:” | “Enter a number to search:” | Y |
| 7 | Option 6  Check | 6 | int | “Question object was created” | “Question object was created” | Y |
| 8 | Option 11  Check | 11 | int | “Press any key to continue” and system exit | “Press any key to continue” and system exit | Y |
| 9 | Option 8 Number 1 Check | a | char | “Not a number” | “Not a number” | Y |
| 10 | Option 8  Number 2 Check | a | char | “Not a number” | “Not a number” | Y |
| 11 | Option 8 Operator Check 1 | a | Char | “Not an operator” | “Not an operator” | Y |
| 12 | Option 8  Operator Check 2 | 8 | int | “Not an operator” | “Not an operator” | Y |
| 12 | Option 7  Check | 7 | int | Console background turns to blue | Console background turns to blue | Y |
| 13 | Option 9 Play music | 9 | int | Music starts playing | Music starts playing | Y |
| 14 | Option 10  Stop music | 10 | int | Music stops playing | Music stops playing | Y |
| 15 | Word search check | “zzz” | string | showMenu() | showMenu() | Y |
| 16 | Option 5 Search Check 1 | “a” | int | “Did not find!” | “Did not find!” | Y |
| 17 | Option 5  Search Check 2 | “1” | int | “1 exists!” | “1 exists!” | Y |

# Screenshots

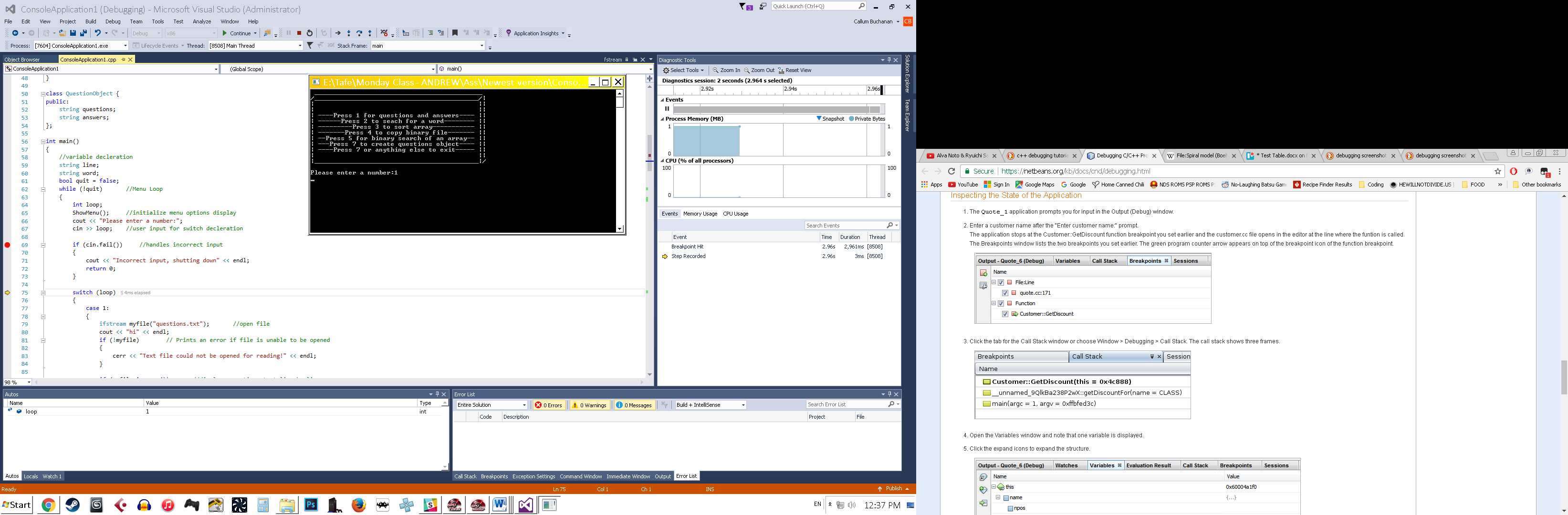
#1 Upload to GitHub

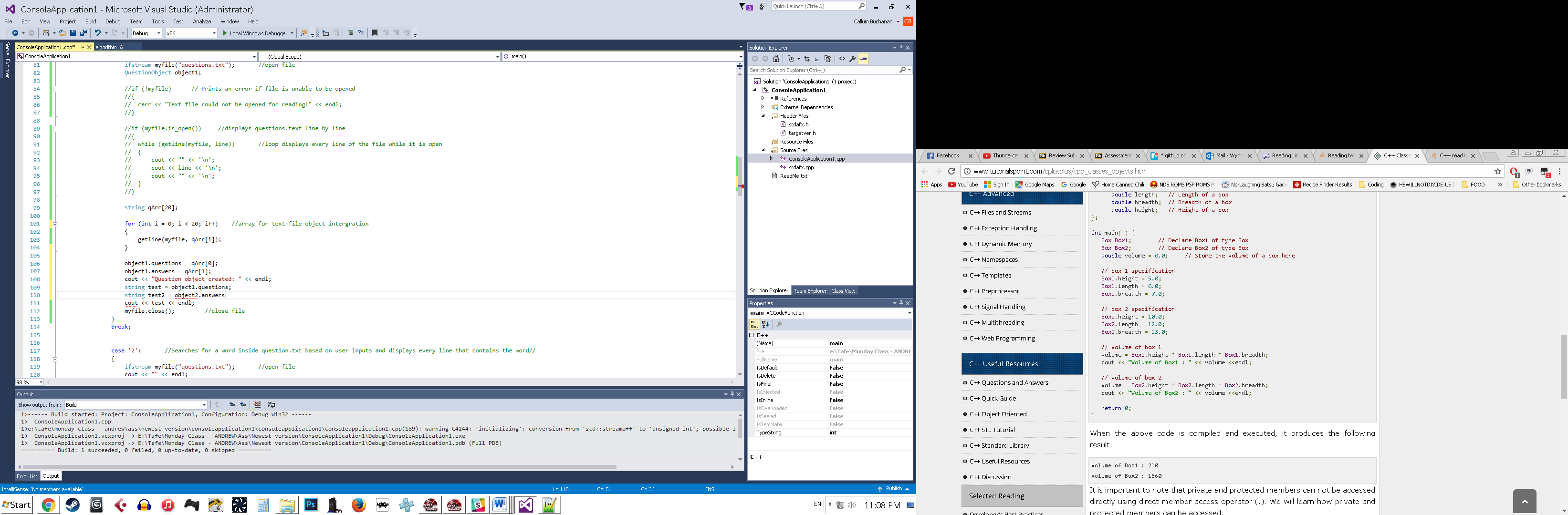


#2 Testing word search:



#3 Testing Menu Loop:



#4 Sample from incomplete code:

# Automated Build Script

This automated build script can be used in the software called “Cmake” to build a C++ Visual Studio executable:

1 cmake\_minimum\_required(VERSION 2.8.9)

2 project (hello)

3 add\_executable(hello helloworld.cpp)

Line 1: Establishes Cmake version to be used

Line 2: Establish a project name in brackets

Line 3: Tells Cmake to create new exe, named “hello”, using the “helloworld” C++ project

# Client Sign Off

Team Leader: Jack Williams

Team Leader Sign-Off: Jack Williams Date: 13/03/2017

Programmer: Callum Buchanan

Developer Sign-Off : Callum Buchanan Date:13/03/2017

Client: Lucas Arouca

Client Sign-Off: Lucas Arouca Date: 13/03/2017