Advanced Higher Computing Science Project

*James McCorkindale*

A picture containing table

Description automatically generated*SCN: 1010 25152*

Contents

[Software Specification 6](#_Toc99706667)

[Purpose 6](#_Toc99706668)

[Scope 7](#_Toc99706669)

[Software 7](#_Toc99706670)

[Design 7](#_Toc99706671)

[Implementation 7](#_Toc99706672)

[Testing 7](#_Toc99706673)

[Documentation 7](#_Toc99706674)

[Evaluation 7](#_Toc99706675)

[Website 8](#_Toc99706676)

[Design 8](#_Toc99706677)

[Implementation 8](#_Toc99706678)

[Boundaries 9](#_Toc99706679)

[Application Boundaries 9](#_Toc99706680)

[Restrictions 9](#_Toc99706681)

[End-User Requirements 9](#_Toc99706682)

[Functional Requirements 10](#_Toc99706683)

[Inputs 10](#_Toc99706684)

[Processes 11](#_Toc99706685)

[Output 12](#_Toc99706686)

[Constraints 13](#_Toc99706687)

[Technical 13](#_Toc99706688)

[Software Design and Development 13](#_Toc99706689)

[Website Design and Development 13](#_Toc99706690)

[Economical 14](#_Toc99706691)

[Software 14](#_Toc99706692)

[Training 14](#_Toc99706693)

[Timeline 14](#_Toc99706694)

[Further Notes 15](#_Toc99706695)

[UML Use Case Diagram 16](#_Toc99706696)

[Gannt Chart Timeline 18](#_Toc99706697)

[Design 21](#_Toc99706698)

[Software Design 21](#_Toc99706699)

[User Interface/Application Layout 21](#_Toc99706700)

[Class Diagrams 22](#_Toc99706701)

[Pseudo Code 34](#_Toc99706702)

[Web Design 35](#_Toc99706703)

[Hierarchal Diagram 35](#_Toc99706704)

[User Interface 36](#_Toc99706705)

[Implementation 39](#_Toc99706706)

[Code for Software 39](#_Toc99706707)

[DesignArk.h 39](#_Toc99706708)

[DesignArk.cpp 41](#_Toc99706709)

[MainFrm.h 48](#_Toc99706710)

[MainFrm.cpp 50](#_Toc99706711)

[ChildFrm.h 58](#_Toc99706712)

[ChildFrm.cpp 59](#_Toc99706713)

[Resource.h 61](#_Toc99706714)

[pch.h 64](#_Toc99706715)

[pch.cpp 65](#_Toc99706716)

[framework.h 66](#_Toc99706717)

[targetver.h 68](#_Toc99706718)

[CDocView.h 69](#_Toc99706719)

[CDocView.cpp 70](#_Toc99706720)

[CTextDocView.h 71](#_Toc99706721)

[CTextDocView.cpp 74](#_Toc99706722)

[CTextDocument.h 111](#_Toc99706723)

[CTextDocument.cpp 112](#_Toc99706724)

[CTextHandler.h 120](#_Toc99706725)

[CTextHandler.cpp 121](#_Toc99706726)

[CAppObject.h 130](#_Toc99706727)

[CAppObject.cpp 131](#_Toc99706728)

[CTextEditorObject.h 134](#_Toc99706729)

[CTextEditorObject.cpp 137](#_Toc99706730)

[CTextLineObject.h 168](#_Toc99706731)

[CTextLineObject.cpp 170](#_Toc99706732)

[CSmartColour.h 185](#_Toc99706733)

[CSmartColour.cpp 186](#_Toc99706734)

[CSmartColourStatements.h 187](#_Toc99706735)

[CSmartColourStatements.cpp 188](#_Toc99706736)

[CSmartColourTypes.h 191](#_Toc99706737)

[CSmartColourTypes.cpp 192](#_Toc99706738)

[CSmartColourBuiltin.h 194](#_Toc99706739)

[CSmartColourBuiltin.cpp 195](#_Toc99706740)

[CSmartColourOther.h 198](#_Toc99706741)

[CSmartColourOther.cpp 199](#_Toc99706742)

[CAboutDlg.h 201](#_Toc99706743)

[CAboutDlg.cpp 202](#_Toc99706744)

[Code for Website 203](#_Toc99706745)

[index.php 203](#_Toc99706746)

[downloads.php 205](#_Toc99706747)

[userdocs.php 208](#_Toc99706748)

[banner.html 210](#_Toc99706749)

[nav.html 211](#_Toc99706750)

[footer.html 213](#_Toc99706751)

[Final Product 215](#_Toc99706752)

[Application 215](#_Toc99706753)

[Website 216](#_Toc99706754)

[New Skills/Knowledge Evaluation 218](#_Toc99706755)

[Log of On-Going Tests 219](#_Toc99706756)

[Highlighting Bug 219](#_Toc99706757)

[Error When Pressing the Sidebar Button After Highlighting 221](#_Toc99706758)

[About Dialog Assertion Failure 222](#_Toc99706759)

[Copying Nothing Bug 224](#_Toc99706760)

[Testing 225](#_Toc99706761)

[Test Plan 225](#_Toc99706762)

[Persona Testing 229](#_Toc99706763)

[Testing Requirements (Actual Results of Texting) 230](#_Toc99706764)

[Test Case ID 1 230](#_Toc99706765)

[Test Case ID 2 230](#_Toc99706766)

[Test Case ID 3 231](#_Toc99706767)

[Test Case ID 4 232](#_Toc99706768)

[Test Case ID 5 233](#_Toc99706769)

[Test Case ID 6 234](#_Toc99706770)

[Test Case ID 7 235](#_Toc99706771)

[Test Case ID 8 235](#_Toc99706772)

[Test Case ID 9 236](#_Toc99706773)

[Test Case ID 10 236](#_Toc99706774)

[Test Case ID 11 237](#_Toc99706775)

[Test Case ID 12 237](#_Toc99706776)

[Test Case ID 13 237](#_Toc99706777)

[Test Case ID 14 238](#_Toc99706778)

[Test Case ID 15 239](#_Toc99706779)

[Test Case ID 16 239](#_Toc99706780)

[Test Case ID 17 240](#_Toc99706781)

[Test Case ID 18 240](#_Toc99706782)

[Test Case ID 19 241](#_Toc99706783)

[Test Results 242](#_Toc99706784)

[Test Case ID 1 242](#_Toc99706785)

[Test Case ID 2 242](#_Toc99706786)

[Test Case ID 3 242](#_Toc99706787)

[Test Case ID 4 242](#_Toc99706788)

[Test Case ID 5 242](#_Toc99706789)

[Test Case ID 6 242](#_Toc99706790)

[Test Case ID 7 242](#_Toc99706791)

[Test Case ID 8 242](#_Toc99706792)

[Test Case ID 9 242](#_Toc99706793)

[Test Case ID 10 242](#_Toc99706794)

[Test Case ID 11 242](#_Toc99706795)

[Test Case ID 12 242](#_Toc99706796)

[Test Case ID 13 242](#_Toc99706797)

[Test Case ID 14 243](#_Toc99706798)

[Test Case ID 15 243](#_Toc99706799)

[Test Case ID 16 243](#_Toc99706800)

[Test Case ID 17 243](#_Toc99706801)

[Test Case ID 18 243](#_Toc99706802)

[Test Case ID 19 243](#_Toc99706803)

[Evaluation 244](#_Toc99706804)

[Evaluation Report 244](#_Toc99706805)

[Fitness For Purpose 244](#_Toc99706806)

[Maintainability 245](#_Toc99706807)

[Robustness 245](#_Toc99706808)

# Software Specification

## Purpose

The purpose of this program is to allow the end-user to be able to design their software project’s pseudocode, in a Windows Desktop Application, with specialised features. This application will have unique features that will be implemented such as SmartColour which will enable the user to define colours for specific key words depending on their writing style, as well as subline editing capabilities to write pseudo code with sublines.

A website will also be developed with this project. It will be a basic website that will have multiple web pages, allowing the user to view a homepage (which describes the application and displays some of its features), a download page (which allows the user to download the application and the user documentation) and a page which the user may view the user documentation in.

To meet the SQA Advanced Higher Computer Science project requirements, the software will be written with object-orientated programming and will include at-least one array of ‘CString’ objects in each property page of the CSmartColour property sheet (CSmartColourTypes, CSmartColourStatements, CSmartColourBuiltin, CSmartColourOther). Each array is a linked list using the ‘std::vector’ type. These arrays will all be sorted using an insertion sort whenever the user closes its associated property sheet.

## Scope

### Software

#### Design

* A completed design of the software structure includes:
  + A design that precisely meets the end-user requirements
  + Application class structure
  + Pseudocode for the implemented standard algorithm, insertion sort
* A completed design of the user interface that includes:
  + A wireframe diagram of the application, including:
    - Main view
    - Dialog boxes
  + Descriptions that show inputs, processes, and outputs, labelled on the wireframe diagram

#### Implementation

* A working application which allows the user to design pseudo code for software projects
* The application software must precisely match the design
* The application user interface must precisely match the design
* Description of new skills and/or knowledge researched and developed
* A log of on-going testing, including:
  + A description of issues resolved
  + References used to resolve these issues

#### Testing

* A completed test plan for final testing of the fully implemented solution, including:
  + All requirements
  + Description of tests
  + Persona and test cases
* Evidence of requirements testing
* A report on the results of the final testing, in terms of fitness for purpose, maintainability and robustness

#### Documentation

* User documentation that is to be available via the website

#### Evaluation

* A detailed evaluation report in terms of:
  + Fitness for purpose, by discussing:
    - How closely the solution matches the requirements specification
    - The results of the testing
  + Maintainability
  + Robustness

### Website

#### Design

* A design that precisely meets the end-user requirements
* A completed design of the website interface that includes:
  + A hierarchical diagram
  + Wireframe Diagrams
  + Descriptions that show inputs, processes, and outputs, labelled on the wireframe diagram

#### Implementation

* A robust website that allows the user to:
  + Download the software package and install it onto their Windows machine
  + Download user documentation as a PDF
  + View user documentation on the website
* The website must precisely match the design

## Boundaries

### Application Boundaries

#### Restrictions

* The user will only be able to use monospaced fonts

## End-User Requirements

The user will expect to be able to complete the following tasks:

* Create a new file that will allow the end-user to write pseudo code
* In a file, be able to create and develop pseudo code:
  + Write in a text editor with standard functions like highlighting, copy and paste, etc
  + Add sublines for each line
    - Edit them as if they were their own text editor
    - Increment/decrement when the parent line is incremented/decremented
    - Restriction on editing them
      * Cannot remove the line via highlight
      * Cannot create a new line in the middle of the text
  + Zoom in and zoom out of the file
  + For all text be able to do the following:
    - Change fonts (SmartColour)
    - Change colours for set words and add new colours and words (SmartColour)
    - Change the colour of the highlighter
  + Be able to print the files in a slick layout (with the SmartColour settings)
  + Export project as PDF (same as what will be printed)
  + Be able to save and load files

## Functional Requirements

### Inputs

Take user inputs for the following functions. They will either be in the form of buttons, shortcuts, or interaction with the view. Not all functions will be always available.

* Menu Functions
  + File
    - New (text file)
    - Open
    - Close
    - Save
    - Save as
    - Print
    - Print Setup
    - Export as PDF
    - Open a recent file
    - Exit
  + Edit
    - Cut
    - Copy
    - Paste
  + View
    - Toolbars and docking windows
      * Enable/Disable Toolbar
      * Customise Menu bar
    - Enable/Disable status bar
    - Open SmartColour dialog box
      * Edit keywords
      * Edit colours
      * Edit indentation sizes
      * Edit fonts
      * Edit comments type
  + Window
    - New window
    - Manage tabs
  + Help
    - About
    - Help
      * Open Website
      * View user documentation on website
* Tool bar buttons
  + New (text file)
  + Open
  + Save
  + Cut
  + Copy
  + Paste
  + Print
  + About
* Interact with CDocView
  + Text editor/pseudocode editor
    - Create new line in place
    - Create a new subline
    - Backspace function
      * Deletes previous character
      * Deletes Line
      * Deletes highlighted segment
      * Delete subline
    - Interact with caret
      * Arrow keys to move it
      * Click mouse to move it
    - Edit highlighted region
      * Copy
    - Paste clipboard

### Processes

* Menu Functions
  + File
    - Create a file as a tab for editing
    - Open a file as a tab for editing
    - Save files
    - Open OS printing and communicate with OS to print file
    - Edit OS print settings
    - Save a file as pdf (Export as PDF)
    - Exit the application by completing any processes, handling the memory etc, and close the window
  + Edit
    - Copy text to the clipboard
    - Receive text from the clipboard and insert them into the application (paste)
  + View
    - Open a window that enables the customisation of the main window
    - Make the status bar visible and invisible
    - Edit the theme of the main window (colours, styles etc)
    - Open and close any window panels (tabs)
    - Open the SmartColour dialog box
      * Add/remove keywords
      * Change colour for keywords, comment, highlight colour and number colour
      * Change font, indentation size
  + Window
    - Open a new window (run a new instance of Design Ark app)
    - Open a new window to manage open tabs and be able to close them from the window
  + Help
    - Open website, in the default browser
    - Open a window displaying info on the application
* Toolbar Functions
  + Create a file as a tab for editing
  + Open a file as a tab for editing
  + Save files
  + Copy objects to the clipboard
  + Receive objects from the clipboard and insert them into the application
  + Open OS printing and communicate with OS to print file
  + Open a window displaying info on the application
* Interact with CDocView
  + Add/Remove text to lines
  + Add/remove any lines and subline blocks
  + Run any necessary GUI objects such as highlighting and SmartColour

### Output

* Functions
  + Prints to a printer
  + A PDF formatted document with the pseudo code
  + Save as a .txt file, which can be imported back into the app, or edited in another text editor
* CDocViews
  + Text on each line as appropriate in a readable format
  + Text may be coloured depending on the SmartColour formats
  + View will render at an appropriate time, ensuring the windows aren’t rendering when they don’t have to, to make the program more efficient

## Constraints

### Technical

Following is a list showing what packages and software this project will be using during what stages.

##### General

Operating systems used in the project:

* Windows 10
* Windows 11

##### Analysis, Testing and Evaluation

Software that was used to construct the analysis, testing and evaluation:

* app.diagrams.net
* Microsoft Word
* Microsoft Excel
* ilovepdf.com (for merging PDFs together)
* Microsoft Edge
* Google Chrome

#### Software Design and Development

##### Design

Software that was used to construct the software design:

* app.diagrams.net

##### Implementation

IDEs used for software development:

* Microsoft Visual Studio 2019
* Microsoft Visual Studio 2022

Languages used for software development:

* C++
* C

Notable Internal Dependencies (Header Files) used for software development:

*MFC Libraries*

* afxwin.h
* afxext.h
* afxdisp.h
* afxcmn.h
* afxcontrolbars.h

#### Website Design and Development

##### Design

Software that was used to construct the user interface:

* app.diagrams.net

##### Implementation

IDE used for software development:

* Microsoft Visual Studio Code

Languages used for software development:

* HTML
* CSS
* Javascript
* PHP

Other Packages Used:

* Apache Server

### Economical

#### Software

* This project meets the requirements for the MS Visual Studio Community license and so there will be no cost associated with purchasing an IDE
* There are no additional costs associated with acquiring the MS 365 software packages used in the analysis and design stage as they have already been purchased by me and my place of education
* To create the development environment for the application, an add-on feature/tool for MS Visual Studio 2019 will needed to be downloaded. This feature will take up around 74mb of data so is a negligible cost of production
* The final package will require to be run on a Windows 10 operating system

#### Training

* I will have to increase background knowledge on Win32 API and Microsoft Foundation Classes (MFC). This could take time out of development but has been factored into the timescale.

#### Timeline

Project Start: 23/09/2021

Deadline: 13/01/2021

Analysis Start: 23/09/2021

Developer Training and Upskilling: 23/09/2021

Design Start: 15/10/2021

Implementation Start: 15/11/2021

Testing Start: 21/02/2022

Documentation Start: 18/03/2022

Evaluation Start: 23/03/2022

*\*For precise timeline view Gannt chart attached*

*\*All development sections are completed concurrently, with a small overlap of training and upskilling, and design*

### Further Notes

##### General

* As of 28/03/2022, the device being used to make the project can upgrade to Windows 11. Everything will be moved over onto the new operating system.

##### Implementation

* As of 08/11/2021, MS VS 2022 is released so from that point forth, the rest of the implementation will be using that IDE and any previously written code will be moved for use with that IDE

## UML Use Case Diagram

## Gannt Chart Timeline

# Design

## Software Design

### User Interface/Application Layout

### Class Diagrams

### Pseudo Code

## Web Design

### Hierarchal Diagram

### User Interface

# Implementation

## Code for Software

### DesignArk.h

### DesignArk.cpp

### MainFrm.h

### MainFrm.cpp

### ChildFrm.h

### ChildFrm.cpp

### Resource.h

### pch.h

### pch.cpp

### framework.h

### targetver.h

### CDocView.h

### CDocView.cpp

### CTextDocView.h

### CTextDocView.cpp

### CTextDocument.h

### CTextDocument.cpp

### CTextHandler.h

### CTextHandler.cpp

### CAppObject.h

### CAppObject.cpp

### CTextEditorObject.h

### CTextEditorObject.cpp

### CTextLineObject.h

### CTextLineObject.cpp

### CSmartColour.h

### CSmartColour.cpp

### CSmartColourStatements.h

### CSmartColourStatements.cpp

### CSmartColourTypes.h

### CSmartColourTypes.cpp

### CSmartColourBuiltin.h

### CSmartColourBuiltin.cpp

### CSmartColourOther.h

### CSmartColourOther.cpp

### CAboutDlg.h

### CAboutDlg.cpp

## Code for Website

### index.php

### downloads.php

### userdocs.php

### banner.html

### nav.html

### footer.html

## Final Product

### Graphical user interface, text, application Description automatically generatedGraphical user interface Description automatically generatedApplication

### Website

*Graphical user interface, text

Description automatically generatedHome page:*

Graphical user interface, application

Description automatically generated*Downloads page:*

*Graphical user interface, text

Description automatically generatedUser docs page:*

## New Skills/Knowledge Evaluation

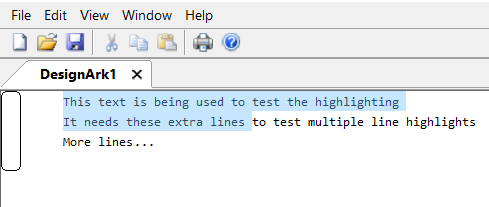
While implementing this project, I have learned many new skills:

* Most notably, I have learned how to effectively use the Microsoft Foundation Classes (MFC) framework, which is the basis of the application that I have made. I have learnt most of the basic skills and architecture needed to create a Windows app, including:
  + The Document/View architecture
  + The MFC Print architecture
  + Using scroll views
  + Making model and modeless dialog boxes
  + Message maps and message handling
  + Using the CRgn class to make more efficient drawing and rendering
  + Using a caret
  + Changing cursor appearances
  + Using the save and open dialog boxes
* I have also gained basic knowledge in using the GDI+ library to have more complex rendering. This was used to make a transparent rectangle for the highlighting and make dynamically sizing bitmap images.
* I have learned the very basics of the Win32 API, however, very minimal amounts of this API were used in the implementation
* I have learned how to use the Windows Clipboard API, to copy/paste text to and from the OS’ clipboard

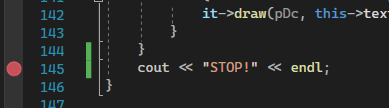
## Log of On-Going Tests

### Highlighting Bug

An error occurs when highlighting a new editor object for the first time. The error causes the highlight to highlight from the start of the editor (line 0, caretPos 0) to the spot the user is currently on.

*Highlighting Glitch:*

To debug this error and to see what’s happening, we should set break points at two places. One when we draw the editor and one when we create the highlight.

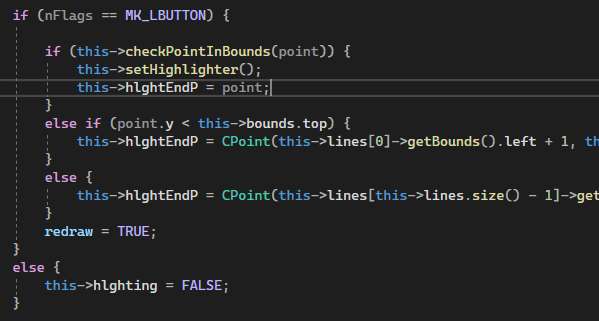
*Breakpoints:*

When we attempt to debug this however, there is a problem. Because we stop the program (and close the app) every time the breakpoint hits, whenever we continue the program, we reopen the app => redraw the app => setting off the breakpoint again, and we are stuck in a loop we can’t break, so it will be impossible to test this. We can fix this by adding a condition on the breakpoint in the draw function; that we only set it off if we are highlighting.

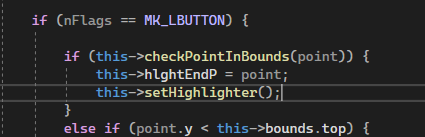
*New Breakpoint:*

Now, when we debug this and see are variable values, we can see that hlghtEndP is set to (0, 0) for the first frame of highlighting. This would then make sense that the highlighting is drawing from there then. Now, we just need to find out where we are setting it to (0, 0), or if we are not setting it in the first place and it is just using its default value.

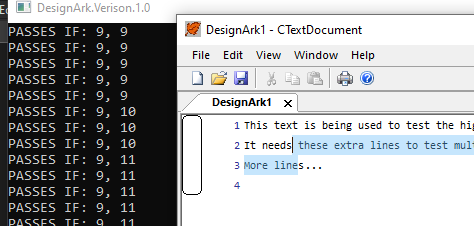
We set hlghtEndP only in the draw function of the object, so we should start by looking there to see if there is anything that doesn’t look right.

*Where we set hlghtEndP:*

It’s not obvious at first, but we are setting the hlghtEndP AFTER we create the highlighter. So, it would make sense that for the first frame, the class sets the highlighter using the default value of the CPoint class (0, 0). So, if we swap the setHighlighter() call and the setting of hlghtEndP, that will probably fix the error.

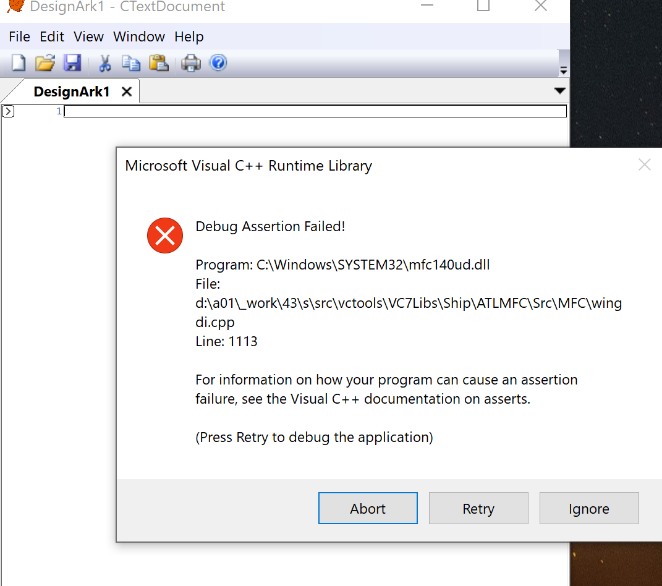
*Fixed code:*

Now when we run that, we can see it works.

*Fixed app:*

### Error When Pressing the Sidebar Button After Highlighting

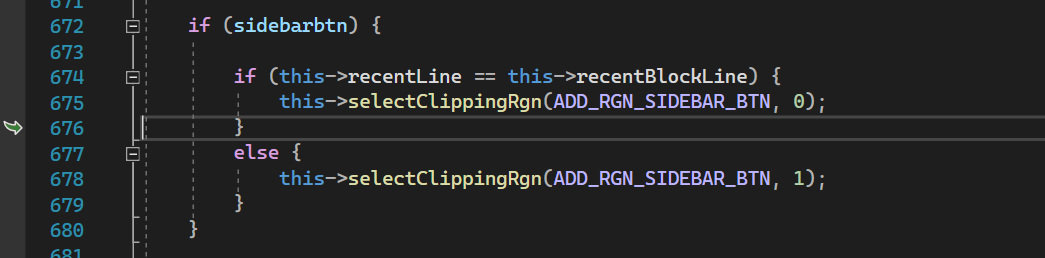
There is a bug that occurs when the user attempts to sidebar button after they have highlighted.

*Initial Failure:*

*Occurring From:*

From this, we can see that the problem is because we are trying to initialise the rgn object despite already having initialising it. We have had this problem before, and the solution was to put in the two different clauses (initialise rgn or combine a CRgn with rgn).

When we trace this thread, we can see that the problem occurs in the previous function (OnLButtonDown), when we decide what value of ‘type’ we will use.

*Not selecting the type to use correctly:*

We can see we are selecting the ‘0’ type at line 675, when we should be selecting the ‘1’ type in line 678.

*More here…*

### About Dialog Assertion Failure

This failure arises from clicking the ‘About’ button and sending the ID\_APP\_ABOUT message to the program. This failure was not always the case as it was working when it was first implemented, so obviously something has been changed since then by accident.

*Graphical user interface, application

Description automatically generatedInitial Failure:*

*Text

Description automatically generatedOccurring From:*

Upon a quick inspection, a problem cannot be easily found. So, at this point, there is two options. 1: To search through the code line by line and possibly find a problem, or 2: Just delete all the code and dialog box and remake everything. The second option will most likely be quicker, and easier, so we will go for that.

After deleting everything, we need to make a new dialog box, using the in-built designer in Visual Studio 2022, we will then make the class and then add the code.

*New Dialog Box:*

*Graphical user interface, application

Description automatically generated*

*Graphical user interface, application, Teams

Description automatically generatedNew Class:*

*Text

Description automatically generatedNew Code:*

We have now implemented a new CAboutDlg (however, it is still identical to the old one), so we need to test it again to see if it works.

*Graphical user interface, text, application

Description automatically generatedAbout Dialog Test:*

It is now working as intended.

### Copying Nothing Bug

While implementing the copying to the clipboard, a bug was noticed that causes an assertion failure when the user attempted to copy (CTRL + C or using a copy button in toolbar/menu) and had not of highlighted anything.

*Graphical user interface, text, application

Description automatically generatedInitial Failure (After attempting to copy):*

*Text

Description automatically generatedOccurring From:*

Text

Description automatically generatedFrom this, we can see that the error occurs due to the variable ‘x’ being void. To fix this, we can simply put a check in the OnCopy function to see if we are highlighting.

*Highlight Check:*

Now, after testing this again, it works and does not create an error.

# Testing

## Test Plan

## Persona Testing

1. Catherine is a teenager studying Higher Computing Science in school. She needs to write pseudo code for her latest coding project and is going to use Design Ark as it is already on her school computers. She has a fair degree of skill using computers, she is fluent in using Windows applications and has had lots of experience in using them.

In general, Catherine found the program easy to use. She enjoys using the subline features which she hasn’t seen in any other apps. However, she has discovered a couple bugs which makes the program not run correctly. Most of the time she must restart the application and loses her work. She says that this might put her off from using the application in the future if it is not fixed soon.

1. Lucy is a senior software engineer. She has worked with computers for her entire career and is well experienced in all sorts of apps and websites. She needs to upgrade the software used on the computers in her office and has chosen to download Design Ark for her employees.

Lucy found the experience of downloading the application on all the computers easy and quick. She enjoys the fact that the storage that the app takes up on the computers is minimal (a few megabytes). However, when downloading the app, a few of the computer’s anti-virus software flagged the download.

When using the app, she likes the simplicity and elegance of the interface. She finds it easy to use and learn. Her favourite part about the app, is the ability to export as PDF, and print the files in a slick format, with easy readability. She also likes the fact that new keywords can be added to be coloured as she tends to write pseudocode in a more unique style than most other developers.

1. Sunny is a graphics designer who has recently starting coding in her free time and wants to learn about pseudocode. She has little to no experience using any technology other than the apps she uses for graphic design.

Despite Sunny’s lack of computer fluency, she enjoys using the website to download the app and to view the documentation. She finds it extremely simple and informative about the application’s capabilities. However, she is mildly put off by the basicness of the website, as a graphic designer, she has come to expect a high level of design in websites.

Sunny enjoys the experience of learning about pseudocode while using the application to write. The SmartColour feature excites her as it adds a bit more of customisation to an otherwise simple app. She likes using it to change the highlight colours and fonts. However, she does admit to this feature distracting her from her learning.

## Testing Requirements (Actual Results of Texting)

### Test Case ID 1

Text

Description automatically generatedGraphical user interface, application, Word

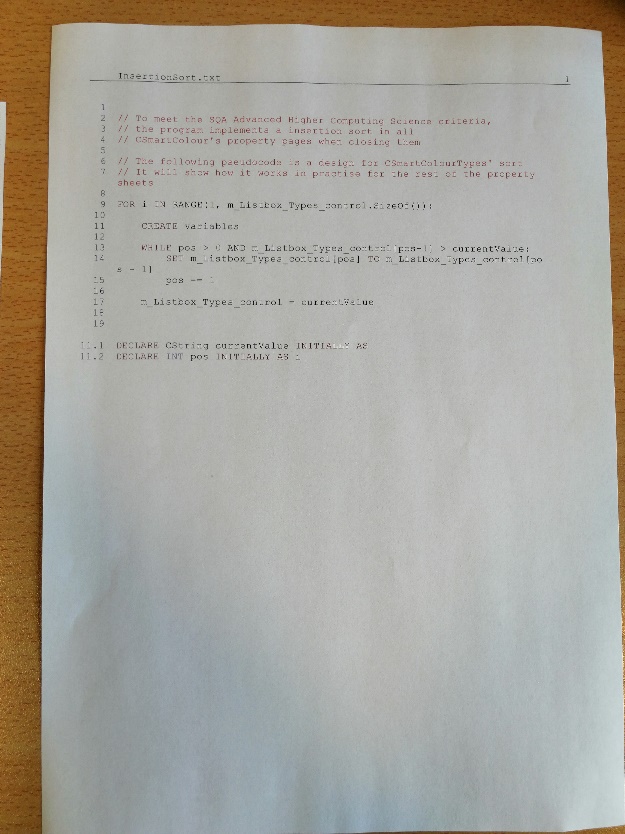
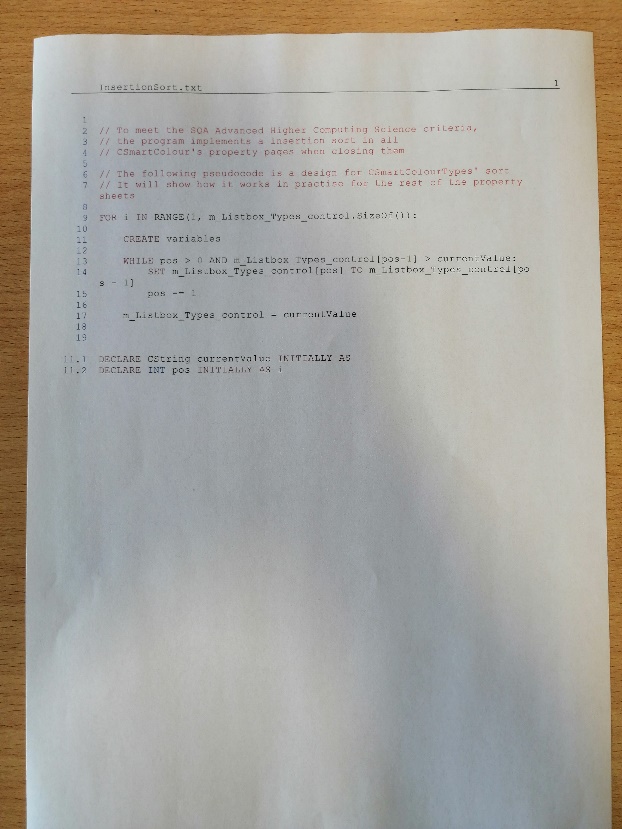
Description automatically generated*Opened new: Opened ‘InsertionSort.txt’:*

### Test Case ID 2

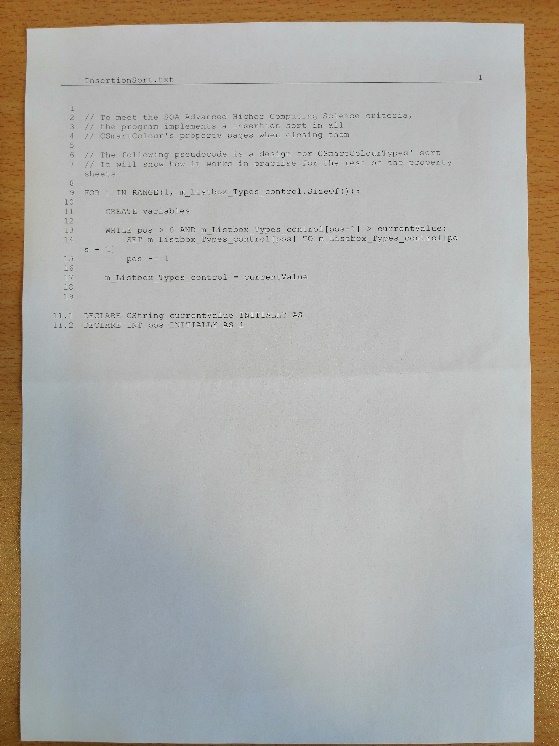
*Text

Description automatically generatedSaved in a .txt file:*

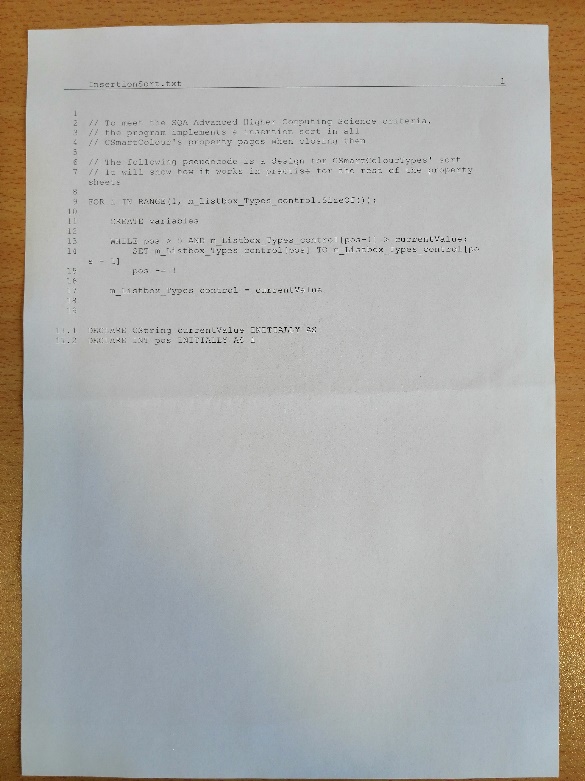
### Test Case ID 3

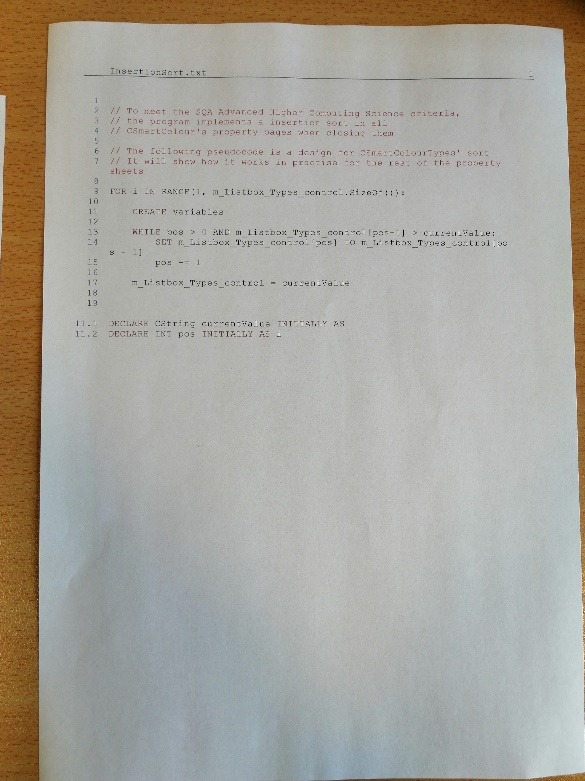
*Home print: School print 1:*

Graphical user interface, text, email

Description automatically generated*School print 2: Exported PDF:*

### Test Case ID 4

*Print with colour: Print with black and white:*

**

### Test Case ID 5

*Graphical user interface, application, Word

Description automatically generatedApplication Open:*

*A screenshot of a computer

Description automatically generated with low confidenceApp on taskbar: App no longer on taskbar:*

*Graphical user interface, text, application

Description automatically generatedApp in OS Task Manager: App no longer in OS Task Manager:*

*Shape

Description automatically generated with medium confidence*

### Test Case ID 6

*Graphical user interface, text, application, email

Description automatically generatedHighlighted and copied text:*

*After copying, pasted text:*

FOR i IN RANGE(1, m\_Listbox\_Types\_control.SizeOf()):

CREATE variables

WHILE pos > 0 AND m\_Listbox\_Types\_control[pos-1] > currentValue:

SET m\_Listbox\_Types\_control[pos] TO m\_Listbox\_Types\_control[pos - 1]

pos -= 1

m\_Listbox\_Types\_control = currentValue

*A picture containing text

Description automatically generatedText in clipboard:*

### Test Case ID 7

A screenshot of a computer

Description automatically generated*Highlighted and copied text:*

*Graphical user interface, text, application, email

Description automatically generatedText pasted into the application:*

### Test Case ID 8

*Graphical user interface, application

Description automatically generatedCustomise window dialog box successfully opening:*

### Test Case ID 9

*Background pattern

Description automatically generatedToolbar enabled:*

*Background pattern

Description automatically generatedToolbar disabled:*

*Graphical user interface, application

Description automatically generatedButton to enable/disable this feature:*

### Test Case ID 10

*Status bar enabled:*

*Status bar disabled:*

*Graphical user interface, application

Description automatically generatedButton to enable/disable this feature:*

### Test Case ID 11

*Some different windows themes:*

*Windows 2000:* *Visual Studio 2005:* Office 2007

*Graphical user interface, application, Word

Description automatically generatedGraphical user interface, application, Word

Description automatically generated* *Graphical user interface, application, Word

Description automatically generated*

### Test Case ID 12

*Background pattern

Description automatically generatedBackground pattern

Description automatically generatedTab closing with two+ tabs already open. Before and after:*

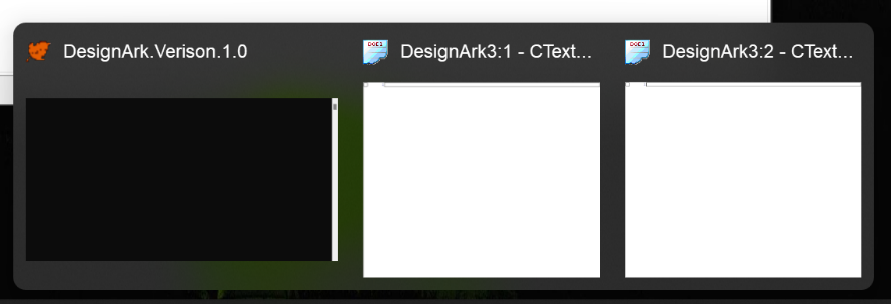
*Graphical user interface, text, application

Description automatically generatedBackground pattern

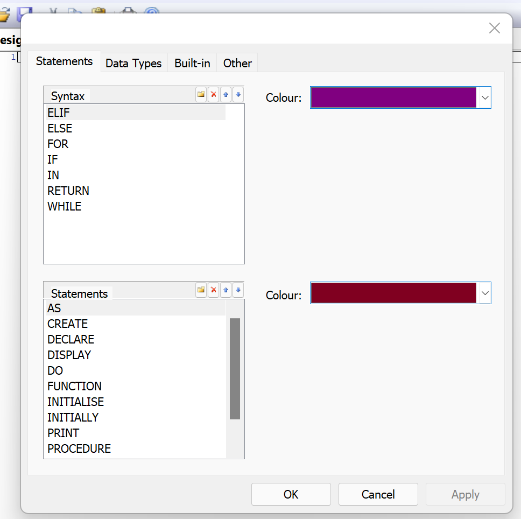
Description automatically generatedTab closing with only one tab already open. Before and after:*

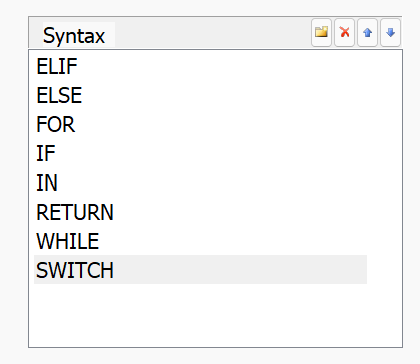
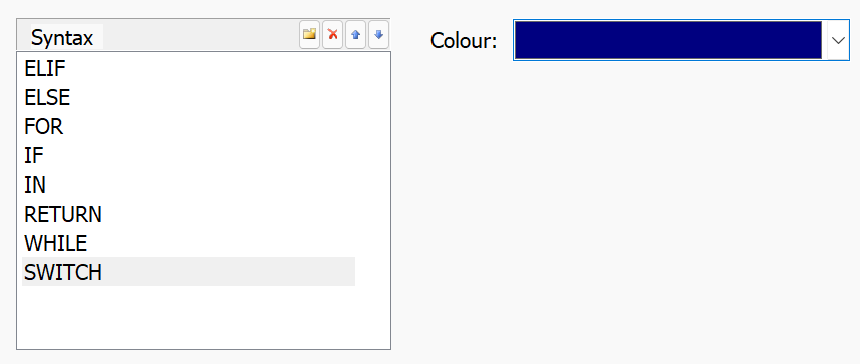
### Test Case ID 13

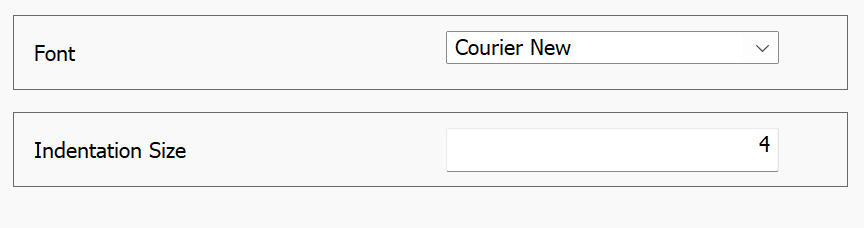
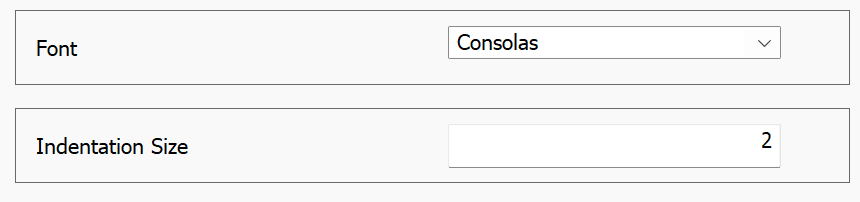
*New window button: New window:*

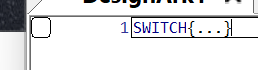
*New window shown on the taskbar:*

### Test Case ID 14

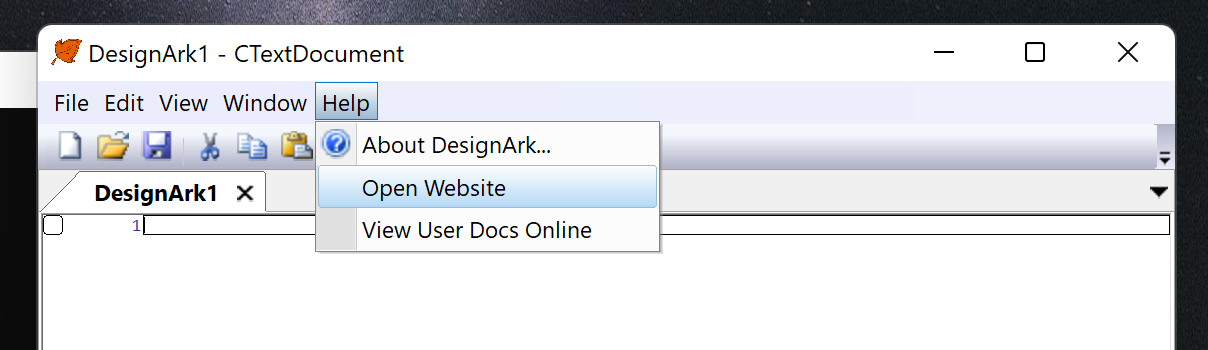
*SmartColour window opening successfully:*

*New keyword can be added:* *Colour can be changed:*

*Fonts and indentation can be changed. Before and after:*

*Changes are updated when SmartColour dialog is close:*

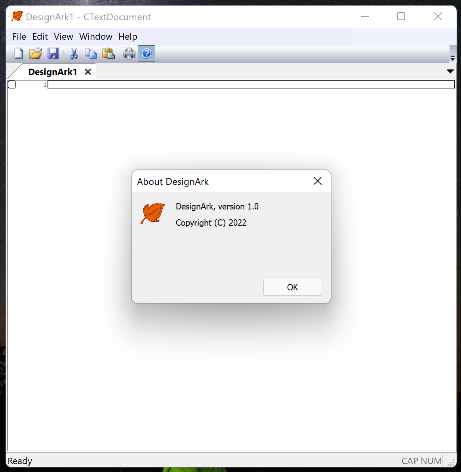
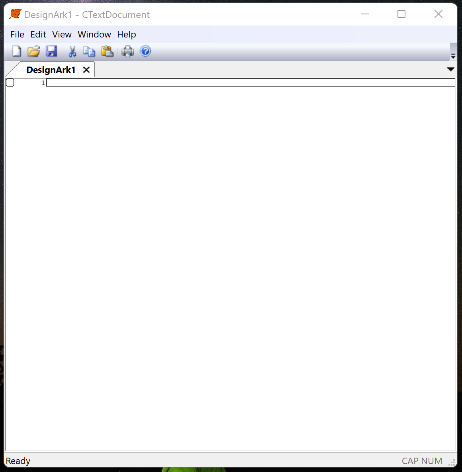
### Test Case ID 15

*Buttons to open the website:*

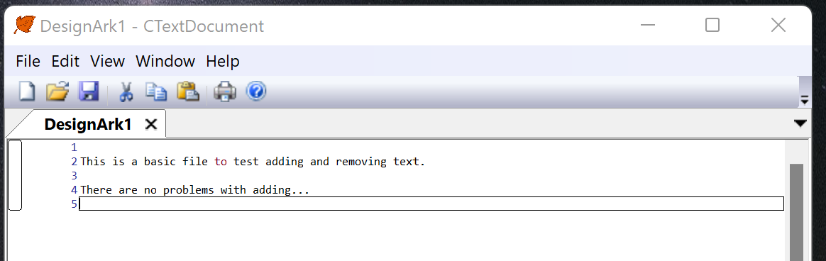
*Graphical user interface, text

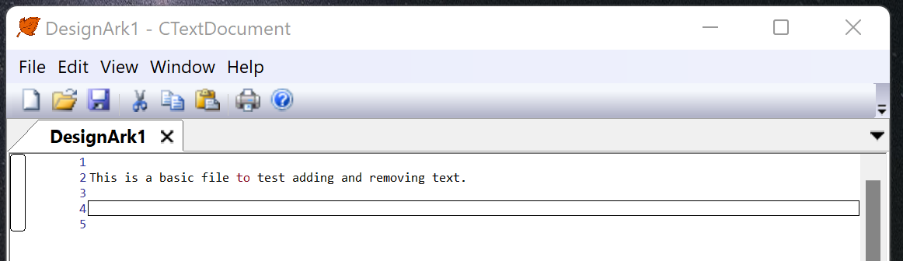
Description automatically generatedOpened website (via buttons):*

### Test Case ID 16

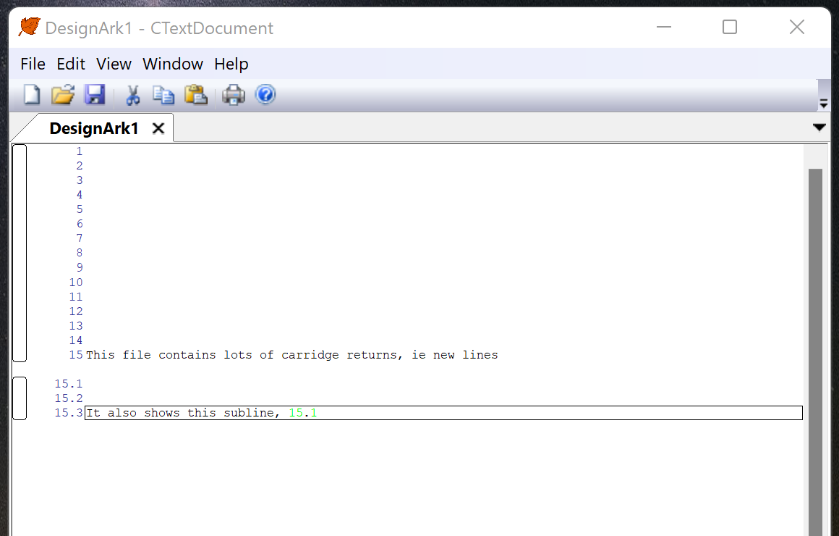
*About dialog. Opens and closes:*

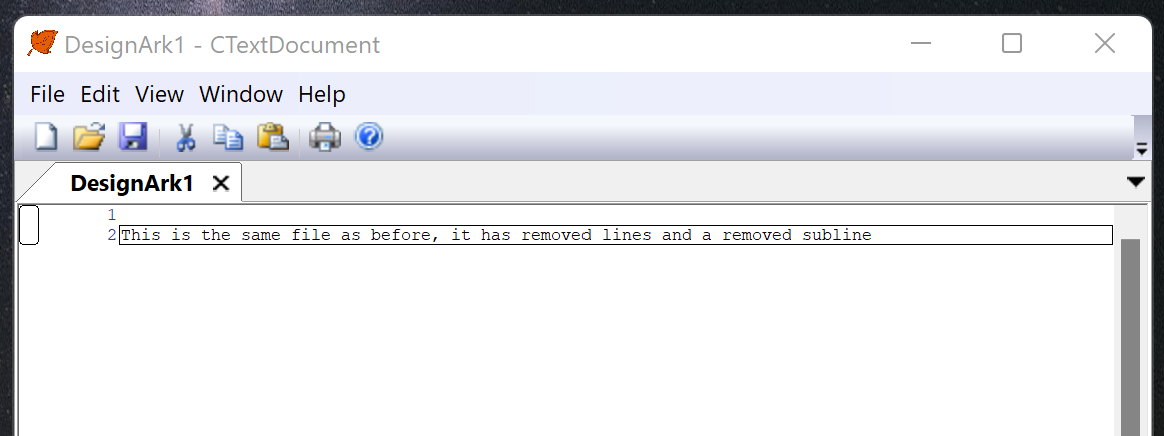
### Test Case ID 17

*Adding text:*

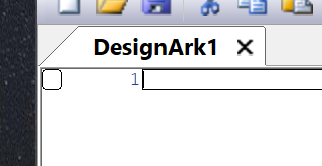
*Removing text:*

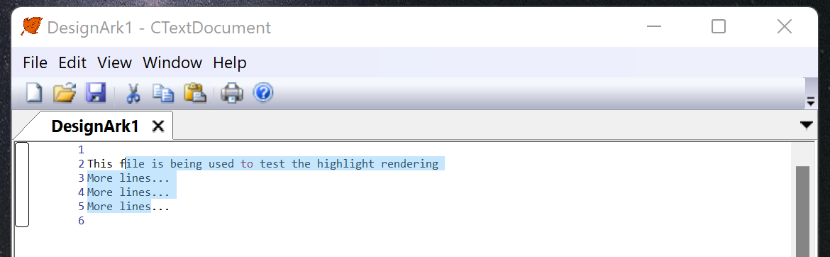
### Test Case ID 18

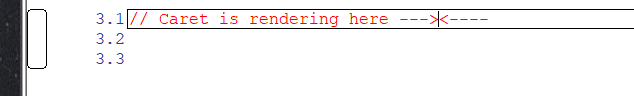
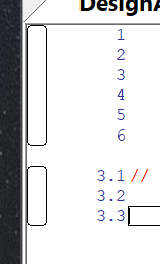
*New lines and subline:*

*Removed lines and deleted sublines (using the same file as before):*

### Test Case ID 19

*Subline arrow button is not rendering properly:*

*Highlight rendering:*

*Sidebar rectangle and line numbers:* *Caret renders:*

Text

Description automatically generated*Coloured text:*

*Cursor type changes at the appropriate place:*

*\*Cursor can not be snipped with MS Snipping Tool, but it is working.*

## Test Results

### Test Case ID 1

Application can successfully open a .txt file in the editor, that contains sublines as well as successfully creating a new, blank file.

### Test Case ID 2

Application can successfully save a document written using the editor, with the line numbers and appropriate spacing.

### Test Case ID 3

Application consistently prints the same file across different printers. It successfully exports to PDF, which matches the print pages.

### Test Case ID 4

Application can effectively change the print settings across different printers.

### Test Case ID 5

Application can close without any errors occurring.

### Test Case ID 6

Application successfully copies the highlighted text to the clipboard. From there, it is able to be pasted into another text editor.

### Test Case ID 7

Application cannot yet successfully show it can paste into itself. When pasting, it will auto-indent the text, even if the indents are already present.

### Test Case ID 8

Application can successfully open the customise window dialog box, to enable the user to customise their app.

### Test Case ID 9

Application can successfully enable and disable the toolbar

### Test Case ID 10

Application can successfully enable and disable the status bar

### Test Case ID 11

Application can successfully change its themes to match the user’s selection

### Test Case ID 12

Application can successfully close tabs. However, no warning occurs if the document has not been saved. This would possibly be something to add in the future.

### Test Case ID 13

Application can successfully instantiate a new window.

### Test Case ID 14

Application can open and close the SmartColour dialog successfully. When it closes it automatically updates the view to show the changes. Keywords, colours, fonts, and indentation sizes can all be successfully changed without issue. However, there is no validation on the indentation size input, so it is possible the user could input an extremely high value, which would give them trouble when writing.

### Test Case ID 15

Application can successfully open the website from the menu buttons.

### Test Case ID 16

Application can successfully open and close the about dialog box.

### Test Case ID 17

Application can successfully add and remove text in the selected editor line.

### Test Case ID 18

Application can successfully add and remove lines and sublines. It can remove lines via highlight and backspace, and there is validation when removing them; they can’t have a subline. When lines are incremented or decremented (usually via lines above or below, being added or removed), their sublines are also incremented or decremented successfully.

### Test Case ID 19

In terms of the GUI, the highlighter, sidebar rectangle, line numbers, caret, coloured text, and cursor all successfully render. The sidebar arrow button, that is used to create a new subline does not render correctly, when the application is not debugging. This is due to it needing access to the bitmap images, which has not been implemented.

# Evaluation

## Evaluation Report

### Fitness For Purpose

The program is fit for purpose. It meets all the end-user requirements, and the functional requirements have all been tested and work, bar a few minor bugs. The program allows the user to write pseudocode in a simple and easy to understand environment.

The user can create a new file for editing, either by opening a new app, or clicking one of the ‘New’ buttons in the app. The app has basic text editor functionalities like copy, paste, cut, and can highlight text. There is a slight bug when pasting text, which causes the lines to gain indentation, however, this can easily be fixed and further versions.

It allows the user to edit sublines in the same view. They work as if they’re their own text editor. The sublines can be added simply by clicking a button on the sidebar, next to the parent line. Sublines can be removed, when they are empty, by clicking the backspace button. Sublines will be incremented/decremented when their parent line is also incremented/decremented. Parent lines will not be able to be removed, via highlight, and a return carriage cannot be added inside the line’s text (i.e., a new line that separates the text).

The app allows the user to zoom in and out of the view, to make the text and other object larger or smaller, depending on the user’s preferences and needs.

It allows the user to add new keywords to the SmartColour feature, add edit any keyword colours. The SmartColour feature also allows highlight colour, font, indentation size and comment colour customisation.

The application has printing capabilities, it can effectively print a well formatted version of a file, using the SmartColour colours. It also can export a PDF format of the document as well, which is identical to the print version.

The application allows the user to save their pseudocode files in a well formatted .txt file. It contains the line numbers and text, with the appropriate spacing. The application can also open .txt files and format them into the editor. It is currently best to only open files that have been created by the application previously, as the application will put any files that don’t contain the sublines’ numbers straight into the first, main editor.

One other bug to note, is the fact that the sidebar button bitmap doesn’t render correctly when the app is not debugging. This doesn’t affect the fitness for purpose as the application is still completely usable, but it may be harder to understand what the button does for first time users.

### Maintainability

The code is extremely maintainable. It contains meaningful variable and class names all the way through the code. This makes it very readable and understandable for anyone else that wishes to edit my project.

There is internal commentary throughout the code. It is used in every algorithm which is particularly hard to read and understand at first glance.

The code is created in modules which will allow very easy addition of new features, for example, a possible future version could contain a graph creator to start integrating all software design onto one platform. The modules also allow for easy bug fixes as every section of the code is limited to a very specific area, like a class or method.

### Robustness

The program is very robust. It allows all text to be written on the editors and new lines can be easily added and removed. The one problem might occur with the indentation size integer input, which does not have any validation and can cause unwanted indentation sizes if the user were to accidentally, or deliberately enter a large number.