SIT323 Practical Software Development, Trimester 2, 2020

Week 3 – Practical 2

GUI and Validation

Introduction

This practical requires students to start creating a GUI, some classes, and validation in Visual Studio.

Tasks

For each TAFF or CFF test file for Assessment Task 1, validity is presented in Table 1.

ID	Allocations File (.taff)	Configuration file (.cff)	Allocations
Test1	valid	valid	valid
Test2	valid	valid	valid
Test3	valid	valid	invalid
Test4	invalid	invalid	NA

Table 1.

1. Attempt to visually identify all errors in Test4.taff and Test4.cff.

    - 21 errors are in Test4.taff
    - 2 errors are in Test4.cff

2. Create a GUI of several forms.

    - There is no need to write any code for this task.
    - At the end of this task, your Solution Explorer might resemble the one depicted in Figure 1.
    - Your GUI might consist of 3 forms as depicted in Figure 2: a main form, a form to display error messages, and a form for an About Box.
    - Menu items of the main form are described below.

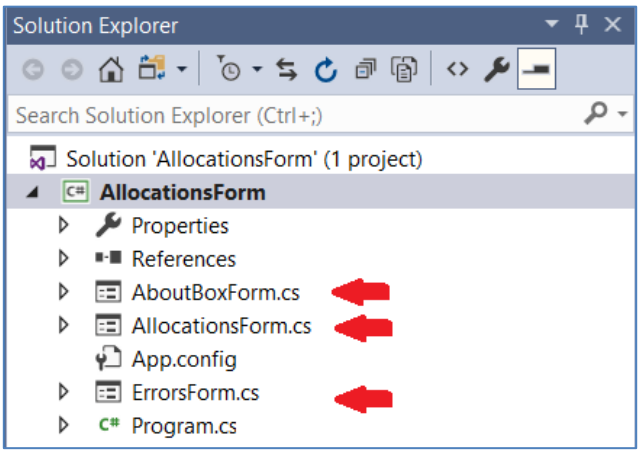


Figure 1.

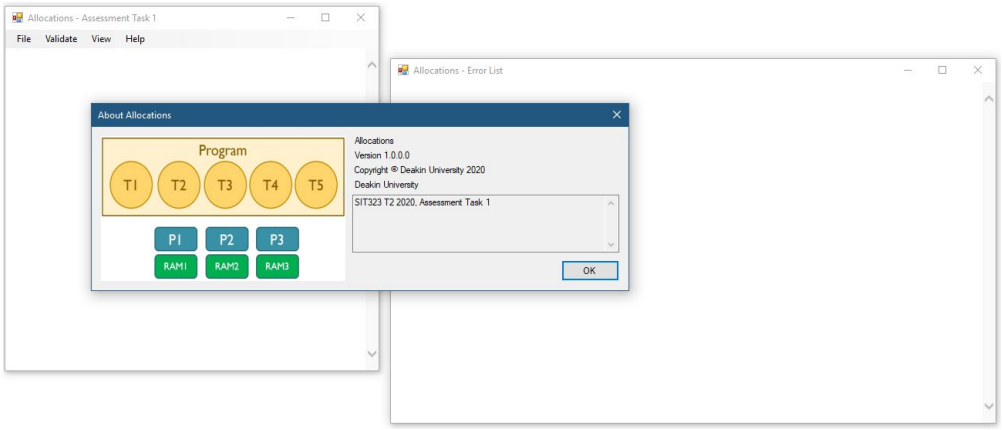
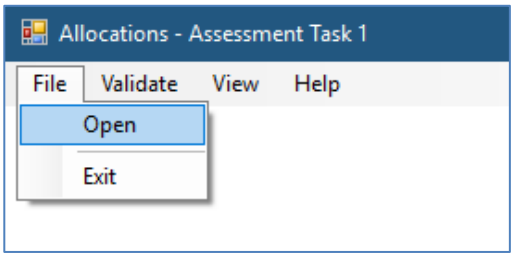
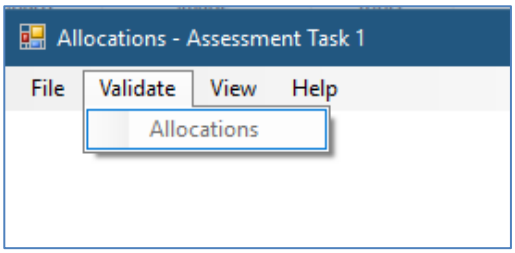


Figure 2.



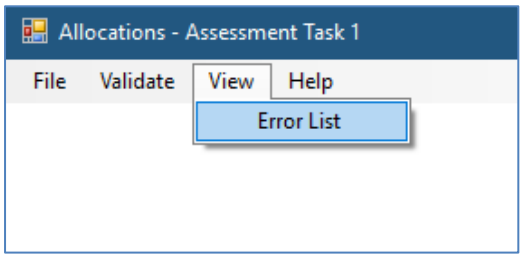
The purpose of selecting the “Open” menu item is to:

- obtain the TAFF filename,
- obtain the CFF filename from the TAFF file,
- validate both CFF and TAFF files,
- load CFF and TAFF data into the program, and
- display data on the GUI.



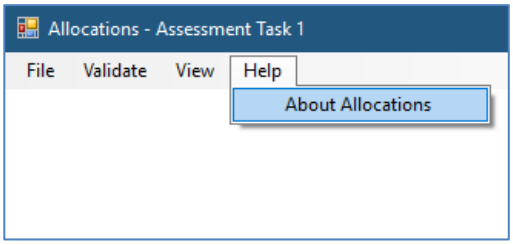
The purpose of selecting the “Allocations” menu item is to apply additional validations to the allocations that were loaded from the TAFF file.

- The “Allocations” menu item starts as disabled, which can be set in the Properties window.
- It is enabled after both TAFF and CFF files are valid.



Each error detected during validation can be stored in some type of collection.

All errors in that collection can be displayed on another window.



An About Box can be displayed.

3. Consider some classes.

   At the end of this task, your Solution Explorer might resemble the one depicted in Figure 3.

   Consider creating a class called TaskAllocations which contains code to at least process the TAFF file. It may also have properties to store data extracted from the TAFF file such as a property to store the name of the CFF file, and many more.

   Consider creating another class called Configuration which contains code to at least process the CFF file. It will have properties to store data extracted from a CFF file.

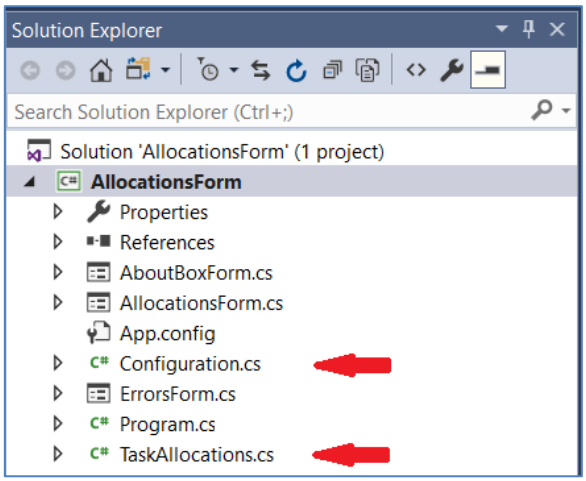


Figure 3

4. One of the first things you program needs to do is obtain the CFF filename that is within the TAFF file.

   Pseudocode to obtain the CFF filename would be similar to the following.

```
OPEN taff file
WHILE NOT EOF
    line = read one line from the taff file
    remove leading and trailing white spaces
    IF line starts with “CONFIG-FILE” THEN
        process this line to obtain the CFF filename
    END IF
END WHILE
CLOSE taff file
```

5. Notice that there are only 4 kinds of lines in a TAFF file:

   i. Blank lines, i.e., lines that contain 0 or more white spaces.
   ii. Lines that contain comments.
   iii. Data lines, each start with a keyword, followed by data on the same line or the next several lines.
   iv. Any other line that doesn't match the above 3 types are simply invalid.

   Pseudocode to process a TAFF file would be similar to the following.

   I recommend that you implement this pseudocode, or similar, in a public method of the TaskAllocations class.

```
OPEN taff file
WHILE NOT EOF
    line = read one line from the taff file
    remove leading and trailing white spaces
    IF line length = 0 THEN
        do nothing as it's just a blank line
    ELSEIF line starts with // THEN
        do nothing as it's just a comment line
    ELSEIF line starts with keyword-1 THEN
        process line(s) with respect to keyword-1
    ELSEIF line starts with keyword-2 THEN
        process line(s) with respect to keyword-2
    ELSEIF line starts with keyword-3 THEN
        process line(s) with respect to keyword-3
    ...

    ELSEIF line starts with keyword-N THEN
        process line(s) with respect to keyword-N
    ELSE
        ERROR invalid data line
    END IF
END WHILE
CLOSE taff file
```

6. Use the pseudocode of task 5, or similar, to implement a similar public method in the Configuration class.

7. Create an event handler for the ***File → Open*** menu-item to:

   i. Obtain a TAFF filename using an OpenFileDialog.
   ii. Create a TaskAllocation object.
   iii. Obtain the CFF filename using the TaskAllocation object.
   iv. Create a Configuration object.
   v. Call the CFF validation method that you created.
   vi. Call the TAFF validation method that you created.