|  |
| --- |
| Software Development Unit 3 Outcome 1Programming Practice School Assessed Coursework (SAC) Part 1 |

**Outcome statement**

*On completion of this unit, the student should be able to interpret designs and apply a range of functions and techniques using a programming language to develop working modules.*

**Task Conditions**

**Allowed resources:** Teacher-provided designs, open book

**Time allocated to this task:** 1 period

**Marks allocated:** 14  
**Location of files:** All files to be submitted for marking MUST be on S:/senior School/Computing/Software Development. Under you own name in an appropriate folder structure, given that you will be undertaking 4 parts to this SAC.

**Task Outline**

Using the module requirements and provided designs, students are required to produce working software modules. They are also required to test each of their modules to ensure that they function correctly.

**Task Summary**

You are required to produce a folio of modules, using only the provided requirements and designs.

You will need to debug and test the modules and their associated applications to ensure they function as expected and are free of errors, using the test table provided.

The marking scheme can be found on the last page.

**Task 1 – Yard duty Swap**

Mr Hunter is in charge of managing the yard duties for staff. He produces a timetable for all staff showing the day and time of their duty and their yard duty area. Eg.

P. Pike: Day1 Recess, L Block

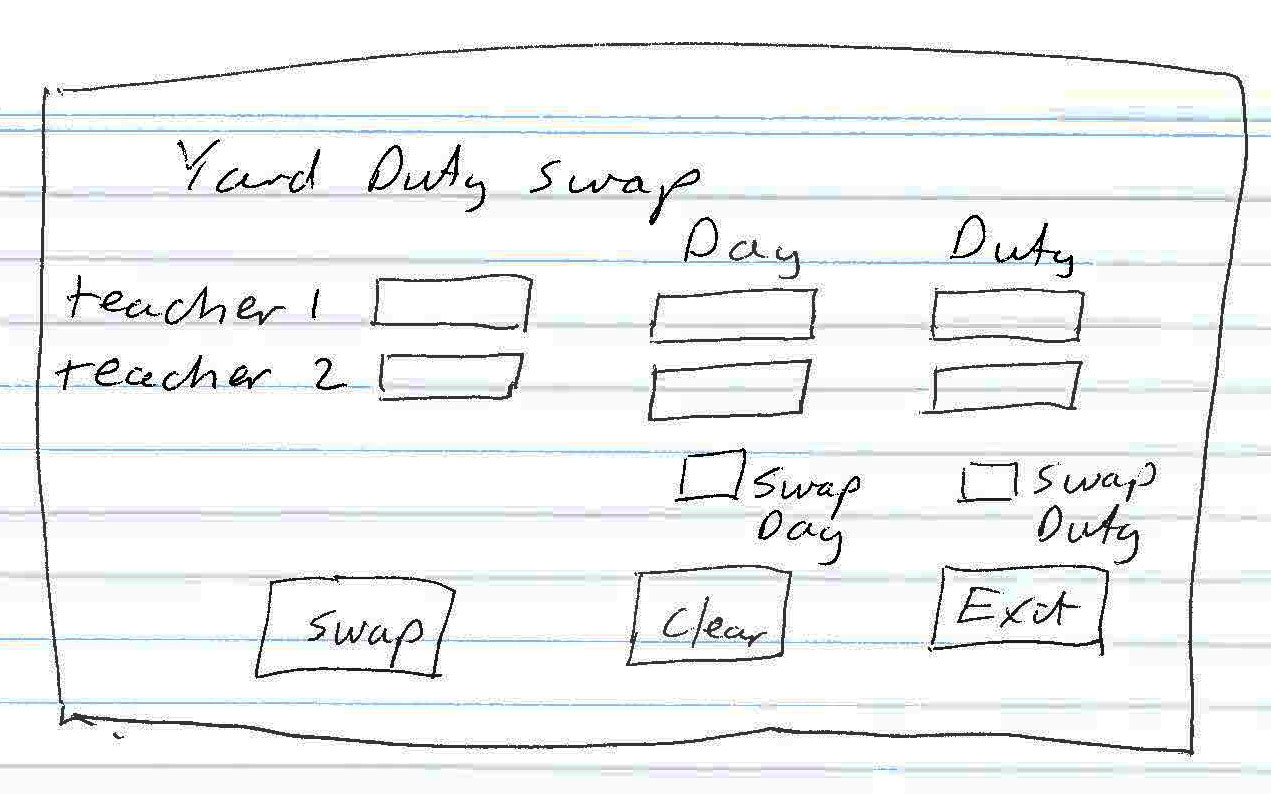
K. Currie: Day3 Lunch, Oval

W. Green: Day2 Before School, Buses

Etc.

Staff are allowed to swap yard duties with each other, but they have to let Mr Hunter know of the swap so he can make the change on his master program. You are required to write a module which will allow Mr hunter to pull up two staff members and their duties from a file and press a button to swap their yard duties. In this program you do not have to store or read the details from a file, simply enter them on the GUI to simulate reading them from a file. The module should swap the days, the duties or both and display the changed duties

A mock-up of the proposed screen is shown below.



**Module** Requirements**:**

Given a two items, the module swaps the contents of each yard duty item

**Provided design: Pseudocode**

Procedure Swap\_duty (itemA, itemB)

temp ← itemA

itemA ← itemB

itemB ← temp

End procedure

|  |
| --- |
|  |

**Testing table**Students are required to test their modules to ensure that they function as described. The testing table should be placed in a separate document called testing.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Item | Test data | Expected Result | Actual result |
|  |  |  |  |
|  |  |  |  |

Marking Scheme

Each task will be assessed using the provided performance descriptors and table below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | **Not shown** | **Low** | | **Medium** | **High** |
| Interpretation of designs to produce working modules | | 0 | 1 | | 2 | 4 |
| Data types and structures used | | 0 | - | | 1 | 2 |
| Processing features used | | 0 |  | | 1 | 2 |
| Testing table | | 0 |  | | 1 | 2 |
| Comments | | 0 | - | | 1 | 2 |
| Subtotal | | 0 |  | |  |  |
|  |
| Total | | /12 |  |