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

**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:** 2 periods

**Marks allocated:** 23

**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.

**Tasks**

**Task 3 – User-defined objects**

At the recent house swimming sports, it was decided that there needs to be a better way to records participants in swimming events and list competitors for each event to give to the starters marshal.

**Module requirements:** This module will facilitate the creation of a user-defined object ( Record structure), which will include five properties and three methods/functions/procedures. You will need to store up to 8 swimmers in each event.

**Provided design: Object description**

|  |  |
| --- | --- |
| **Name:** Student | **Description** |
| **Properties:**  strGivenName  strSurname  chrGender  intYearLevel  StrEvent | Given name of the student  Surname of the student  Gender of the student, either ‘M’ or ‘F’  Year Level number of the student. (7-12)  The event the student is swimming in. (See list below) |
| **Methods/Functions/ Procedures**  AddStudent  EventList  Clear | Adds a student to a swimming event  Enter an event and list all participants in that list  Clears the list of previous competitors |

Events include Freestyle, backstroke, butterfly and breaststroke.

**Processes**

Before the swimming sports, the head of sport enters all the competitors on the input screen. These details are saved to file (not required in this module).

At the swimming sports, the data file is retrieved (not required) and the starters marshal selects the event and a list of competitors is displayed. They are checked and sent to the starting blocks. The final program will select event, gender and year level, but your module only needs to select the event.

**Your task.**

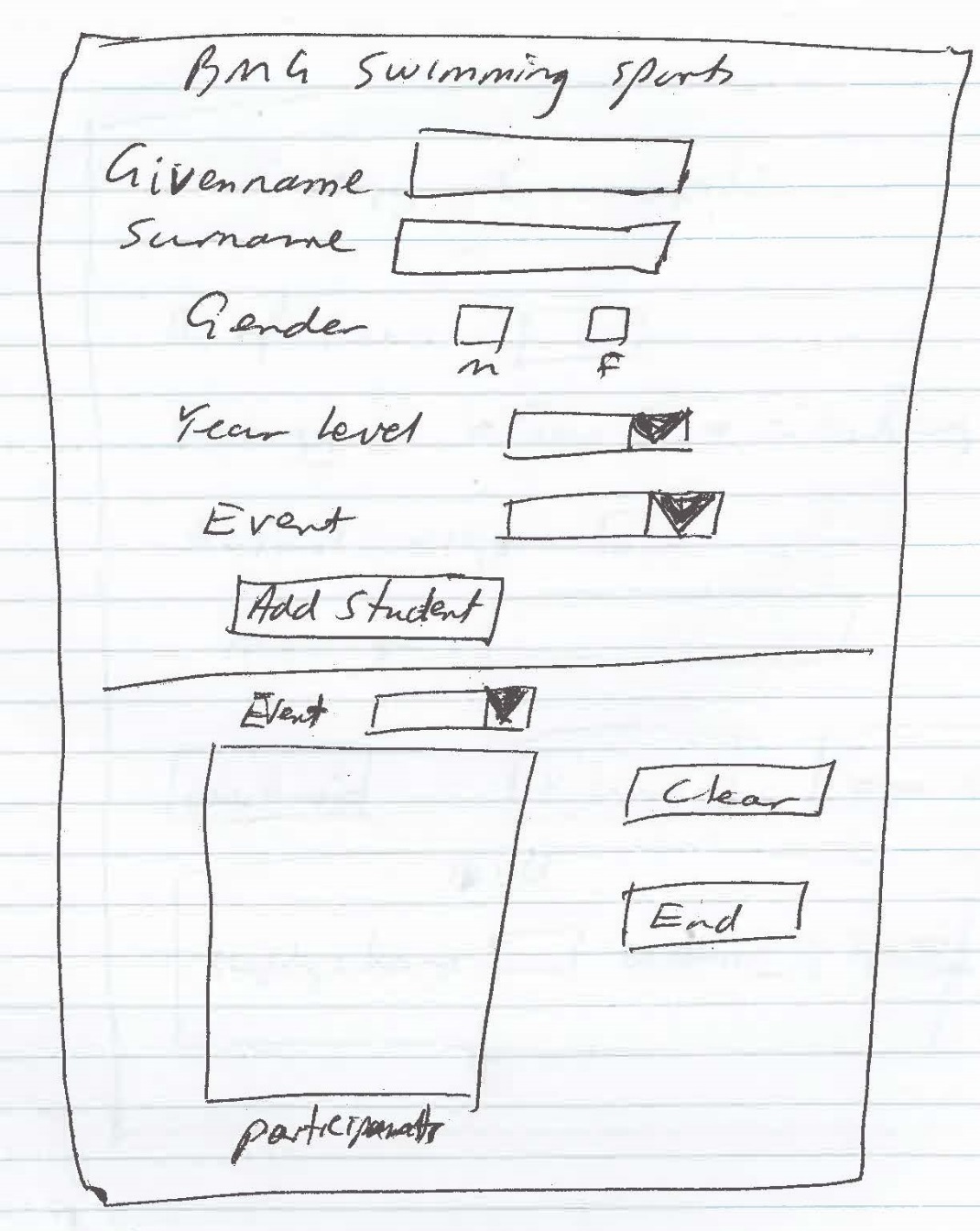
You are to prepare the data entry module and the display module and test your solution.

After each student is added, the input areas should be cleared.

**Extension.**

For 1 mark, find a way to select the event, the gender and year level and only display the competitors in that specific event. Eg girls, year 7, Freestyle.

Provided design: User interface



Search algorithm

Start

Enter event to be searched

Loop through list of students

Add 1 to search\_counter

If student.event( search\_counter) = entered event then

Display student full name, gender and year level

Endif

End Loop

End

Testing table

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Test Data | Expected output | Actual output |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Marking Scheme

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | **Not shown** | **Low** | | **Medium** | **High** |
| Interpretation of provided designs to produce working modules | | 0 | 2 | | 5 | 8 |
| Data types and structures used | | 0 | 2 | | 3 | 4 |
| Processing features used | | 0 | 0 | | 1 | 2 |
| Test data and testing table produced | | 0 | 1 | | 2 | 3 |
| Internal documentation evident | | 0 | 1 | | 2 | 3 |
| Subtotal | |  |  | |  |  |
|  |
| Total | | /20 |  |