|  |
| --- |
| Software Development Unit 3 Outcome 1Programming Folio School Assessed Coursework (SAC) Part 4 |

## **This SAC will consist of 4 Folio Tasks over an extended period of time.**

## U3 SAC 1:  Programming Folio (Contributes 10% of subject assessment)

## **Part 1:  T1W6 FRI 5 March (Double)**

## **Part 2:  T1W8 FRI 19 March (Double)**

## **Part 3:  T2W2 FRI 30 April (Double)**

## **Part 4:  T2W4 FRI 14 May (Double)**

**Outcome 1 statement**

On completion of this unit the student should be able to interpret teacher-provided solution requirements and designs, and apply a range of functions and techniques using a programming language to develop and test working software modules. (study design 2020-2024)

**Task Conditions**

1. **Allowed resources:** Teacher-provided solution designs, open book
2. **Time allocated to this task:** 2 periods (Double) Friday 14 May. **100 minutes**
3. **Marks allocated:** 33  
   **Location of files:** The zipped Visual Basic Project folder is to be submitted for marking Under:
4. **Submission of work: Learning Tasks:** “**SD Unit 3 Outcome 1 Programming Folio Part #4**

**Mr Mac’s Magnificent Motel**

|  |
| --- |
| **713 Vintage Neon Motel Sign Photos - Free & Royalty-Free Stock Photos from  DreamstimeMr Mac’s Magnificent Motel** is a newly established family business in the leafy suburbs of Macville. Mr Mac has requested that you create a user interface that allows for the registering (storing in an external file) and displaying of customer detail. |

**Functional Requirements**

|  |  |
| --- | --- |
| **FR** | **Description** |
| **FR01** | Can list all customers’ details in the Customer File |
| **FR02** | Can Add a new customer to the Customer File |
| **FR03** | Can display the total number of customers in the Customer File |
| **FR04** | Can display a list of names and number of customers in a user-specified “rating” [Bronze, Silver, Gold, Platinum] For example, if the user inputs/selects “Silver” the solution will display a list of customer names that have a Silver rating along with the total number of Silver rated customers. |

**Non-Functional Requirements**

|  |  |
| --- | --- |
| **FR** | **Description** |
| **NFR01** | The application form size must fit the dimensions of a laptop |
| **NFR02** | The user interface must be intuitive for the user |
| **NFR03** | Font styles and colours must present a professional appearance |

**Constraints**

|  |  |
| --- | --- |
| **FR** | **Description** |
| **C01** | The solution must be produced on the School laptop |
| **C02** | You must muse an XML file format for the Customer File |
| **C03** | You must use a Selection Sort or a Quick Sort algorithm |
| **C04** | You must use the Visual Basic .NET programming language |
| **C05** | You must complete the task within 100 minutes. |

**XML Customer File Structure**

Customers are rated as Bronze, Silver, Gold, or Platinum based on their use/frequency of the hotel.

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Records xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<CUSTOMER>

<GIVEN>Todd</ GIVEN >

<SURNAME>Gakk</ SURNAME >

<STREET>45 Smith Avenue</ STREET >

<SUBURB>Melbourne</ SUBURB >

<POSTCODE>3000</ POSTCODE >

<MOBILE>0412999999</ MOBILE >

<RATING>Platinum</ RATING >

</ CUSTOMER >

</Records>

**Testing Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test no.** | **Test Item** | **Test Data** | **Expected results** | **Actual results** |
| 1 | Can list all customers’ details in the Customer File | XML items already added | Listboxes: Displays all customers | As Expected |
| 2 | Can Add a new customer to the Customer File | Given: Thomas  Surname: Tran  Street: Hunter  Suburb: Richmond  Postcode: 3121  Mobile: 420253673  Rating: Gold | Listboxes:  Given: Thomas  Surname: Tran  Street: Hunter  Suburb: Richmond  Postcode: 3121  Mobile: 420253673  Rating: Gold | As Expected |
| 3 | Can display the total number of customers in the Customer File | XML file with 5 customers | MsgBox: “The total number of customer is 5” | As Expected |
| 4 | Can display a list of names and number of customers in a user-specified “rating” | cbxRating: Gold | Listboxes: Display only gold rating customers | Not As Expected – results vary – works sometimes randomly |
| 5 | Validates (e.g. existence) | Textboxes All empty | MsgBox: Error: “Enter a given name” | As Expected |

|  |  |  |
| --- | --- | --- |
| **Test no.** | **Test Item** | **Snip** |
| 1 | Can list all customers’ details in the Customer File |  |
| 2 | Can Add a new customer to the Customer File |  |
| 3 | Can display the total number of customers in the Customer File |  |
| 4 | Can display a list of names and number of customers in a user-specified “rating” |  |
| 5 | Validates (e.g. existence) |  |

'AUTHOR: Thomas Khai Tran

'DATE: 14.05.2021

'TITLE: U3O1 Part IV Folio SAC

'JUSTIFICATIONS:

'--Visual Basic .NET framework was suitable given built-in features such as ability to convert string values to integers via Val/CDbl

'--Do...While Loop statements were suitable as it is a control flow statement that can execute a block of code repeatedly until

'--meeting a condition at the end of the block.

'--Select Case statements suitable as allowed for a specific block of code to be executed out of selected combobox options for rating

'--IF...Else statements were suitable as it allowed execution of code given specified criteria was met (TRUE), while

'--another set of code evaluated FALSE. IF statements were used in conjunction with boolean Yes/No to verify validation

'--For Each...Next statements were suitable as it is an iterative, incremental loop statement used to repeat sequence of

'--statements for specific number of circumstances. Was used to read and publish to .xml file for each node

'VALIDATION:

'Textbox inputs validated using type, range, and existence checks

**Marking Scheme**

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

|  |  |
| --- | --- |
| **Criteria / Skills** | **Marks Available** |
| Interpretation of designs to produce working modules | **3** |
| Data types and structures used | **3** |
| **Use** appropriate processing features of a programming language to develop working modules | **13** |
| **justify** appropriate processing features of a programming language to develop working modules | **4** |
| Develop and apply suitable validation techniques | **3** |
| Develop and apply suitable testing and debugging techniques using appropriate test data | **4** |
| document the functioning of modules and the use of processing features through internal documentation | **3** |
| **TOTAL** | **/33** |

VCE Applied Computing: Performance Descriptors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SOFTWARE DEVELOPMENT UNIT 3 OUTCOME 1 - SCHOOL-ASSESSED COURSEWORK** | | | | | |
| **Performance Descriptors - typical performance in each range** | | | | | |
| ***Unit 3***  ***Outcome 1***  ***Interpret teacher-provided solution requirements and designs, and apply a range of functions and techniques using a programming language to develop and test working software modules.*** | **Very low** | **Low** | **Medium** | **High** | **Very high** |
| Limited interpretation of solution requirements and designs to develop working modules. | Some interpretation of solution requirements and designs to develop working modules. | Sound interpretation of solution requirements and designs to develop working modules. | Most solution requirements and designs are interpreted accurately to developing working modules. | All solution requirements and designs are interpreted accurately to developing working modules. |
| Limited selection and use of data types and data structures. | Some selection and use of appropriate data types and data structures. | Sound selection and use of data types and data structures to develop working modules. | Detailed selection of relevant data types and data structures to develop working modules. | Comprehensive selection of relevant data types and data structures to develop working modules. |
| Limited selection and use of processing features of the programming language to develop some working modules. | Some selection and use of appropriate processing features of the programming language to develop some working modules. | Sound selection and use of appropriate processing features of the programming language to develop some working modules. | Most processing features of the programming language have been selected and used to develop all working modules. | Comprehensive selection and use of relevant processing features of the programming language to develop all working modules. |
| Limited explanation of how the selected processing features are used to develop working modules. | Some justification and explanation of how the selected processing features are used to develop working modules. | Sound justification and explanation of how the selection of appropriate processing features are used to develop working modules. | Detailed justification and explanation of how the selection of appropriate processing features of the programming language are used to develop working modules. | Comprehensive justification and explanation of how the selection of appropriate processing features of the programming language are used to develop working modules. |
| Limited data validation techniques are applied to check the reasonableness of some input data. | Some data validation techniques are effectively applied to check the reasonableness of some input data. | Sound use of data validation techniques are effectively applied to check the reasonableness of input data. | Detailed use of relevant data validation techniques are applied to efficiently and effectively check the reasonableness of all input data. | Comprehensive use of relevant data validation techniques are applied efficiently and effectively to check the reasonableness of all input data. |
| Limited range of test data is expressed in a testing table, with incomplete or missing results. | Some testing of test data is expressed in a testing table with actual output stated. | Sound range of testing of test data is expressed in a testing table, with both expected and actual output stated and some evidence of debugging. | Detailed use of test data is expressed in a testing table, with both expected and actual output stated with evidence of debugging. | Comprehensive use of test data is expressed in a testing table, with both expected and actual output stated, and showing detailed evidence of debugging. |
| Limited internal documentation with few comments regarding the use of the selected processing features. | Some internal documentation with comments regarding the functioning of modules and the use selected processing features. | Sound use of internal documentation with comments regarding the functioning of modules and the use of selected processing features. | Most software modules include detailed internal documentation regarding the functioning of modules and use of selected processing features. | All software modules include comprehensive internal documentation regarding the functioning of modules and use of selected processing features. |

KEY to marking scale based on the Outcome contributing 100 marks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Very Low 1–20 | Low 21–40 | Medium 41–60 | High 61–80 | Very High 81–100 |