**Software Development**

**Unit 3 Outcome 1**

**Programming Practice SAC**

**Outcome 1** 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. To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 1.

**Key skills**

• interpret designs to develop working modules that meet these requirements

• use a range of data types and structures

• use appropriate processing features of a programming language

• select and use appropriate techniques to test the functionality of modules

• document the functioning of modules through the use of internal documentation.

**The Project A: Coffee Carousel**

Coffee Carousel is a local café that wants a basic mobile device App that can process coffee orders sent to the store so that they can have the coffee ready for the customer to pay for and collect.

**Functional Requirements**

The mobile App needs to collect the coffee order of a customer, calculate the cost of that order and keep a record of the order in an array so that customers can re-order at the touch of a button.

The solution is required to collect the following input data through a Validation process:

* First name of the Customer
* Coffee Selection
* Milk Selection
* Sugar Option
* Size Selection
* Save the Order

The solution is required to produce the following output:

* A saved a record of the customers’ order for use in the future
* The cost of the order
* A summary of the order displayed on the screen.

The solution needs to calculate the order based on the following:

* Flat White = $2.50, Cappuccino = $3.00, Short Black = $2.00, Latte = $3.00
* Full Cream Milk = 0.00, Skim Milk = $0.50, Soy Milk = $1.00
* Small = 0.00, Regular = $0.50, Large = $1.00
* Sugar (one) = $0.05, Sugar (two) = $0.10

*Example:*

*An order for a Large Cappuccino with Skim Milk and Sugar (one) =*

*$1.00 + $3.00 + $0.50 + $0.05 = $4.55*

**Non-Functional requirements**

As a prototype your program can use the dimensions: 1024 x 1366 point for an iPad Pro. For more information about mobile device dimensions see this link: <https://designcode.io/iosdesign-guidelines>

Within these dimensions the interface design must be intuitive for the user and follow the conventions of all mobile apps.

Font and colour choices must evoke a warm “coffee house” feel.

**Constraints**

You have to use the Visual Basic programming language during class time only on the following dates (Total of 6 hours):

* Wednesday 1 March (double)
* Thursday 2 March
* Friday 3 March
* Wednesday 8 March (double)
* Friday 10 march
* Wednesday 15 March (double)
* Friday 17 March DEADLINE

**Scope**

The program needs to keep only ONE order for the user only. The program does not have to connect to a phone number and be sent through a phone network. Remember it is only a prototype. It does not transfer funds as the customer will pay on pick up.

**Design Requirements provided by Coffee Carousel**

 

This is a mockup/layout design provided by Coffee Carousel. It implies the functionality of the app.

**Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Name** | **Data Type** | **Size** | **Description** |
| StrCustName | String | 15 | This is the first name of the customer |
| dblCoffee | Double (Floating Point) | 6 | Coffee Choice price |
| strCoffee | String | 10 | Coffee Choice Text |
| dblMilk | Double (Floating Point) | 6 | Milk Choice price |
| strMilk | String | 10 | Milk Choice Text |
| dblSize | Double (Floating Point) | 6 | Size Choice price |
| strSize | String | 7 | Size Choice Text |
| dblSugar | Double (Floating Point) | 6 | Sugar Choice price |
| strSugar | String | 7 | Sugar Choice Text |
| StrOrderName | String | 15 | This is the first name of the customer Saved |
| dblOrderCoffee | Double (Floating Point) | 6 | Coffee Choice price saved |
| strOrderCoffee | String | 10 | Coffee Choice Text saved |
| dblOrderMilk | Double (Floating Point) | 6 | Milk Choice price saved |
| strOrderMilk | String | 10 | Milk Choice Textsaved |
| dblOrderSize | Double (Floating Point) | 6 | Size Choice price saved |
| strOrderSize | String | 7 | Size Choice Text saved |
| dblOrderSugar | Double (Floating Point) | 6 | Sugar Choice pricesaved |
| strOrderSugar | String | 7 | Sugar Choice Textsaved |
| dblCost | Double (Floating Point) | 6 | Total cost of order |
| dblOrderCost | Double (Floating Point) | 6 | Total cost of order saved |

**Assessment Deliverables**

1. Create an Algorithm in either pseudocode or a flow chart that designs the solution for the mobile app.
2. Create the Coffee Carousel Mobile App in Visual Basic
3. Complete the TESTING TABLE below for your solution to check ALL possible outputs.

|  |  |  |  |
| --- | --- | --- | --- |
| Item Tested | Test Data | Expected Result | Actual Result |
|  |  |  |  |
|  |  |  |  |

1. Internal documentation that describes the function of each line of code.

**Assessment Criteria: PART A**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Performance Descriptor** | **Not Included(0)** | **Low(1)** | **Medium(2)** | **High (3)** |  **Comments**  |
| **Variable & Object Names include appropriate type & purpose.**  | *None include appropriate type & purpose.* | *Few include appropriate type & purpose.* | *Some include appropriate type & purpose.* | *All include appropriate type & purpose.* |  |
| **Appropriate Control structures have been employed & correctly implemented.** | *Not a correct control structure for this task and doesn’t work.* | *Not a correct control structure for this task but has created a clumsy solution.* | *Correct control structure used but not well executed* | *All correct* |  |
| **Calculations** | *Not correct* | *Major issues with calculations* | *Minor issues with calculations* | *All correct* |  |
| **Clear Output** | *Not correct and unclear* | *Major issue with output* | *Minor issue with output* | *All output is clear and correct* |  |
| **Internal Documentation** | *Not included* | *Poor attempt at describing code.* | *Moderate attempt at describing lines of code.* | *Thoroughly documents all lines of codes* |  |
| **Interface Design** | *Very Poor* | *Major issues with Design of forms* | *Minor issues with design of forms* | *Excellent form Design* |  |
| **Algorithm** | *Not included* | *Included but there are many issue with the pseudocode or flowchart components and solves the problem* | *Mostly uses pseudocode or flowchart components and solves the problem* | *Correctly uses pseudocode or flowchart components and solves the problem* |  |
| **All functional and non-functional requirements are met** | *No requirements presented*  | *Few requirements presented* | *Most requirements presented* | *All requirements presented* |  |
| **Layout Design followed and improved upon** | *Not followed* | *Followed but not improved upon.* | *Followed with minor improvements* | *Followed with extensive improvements.* |  |
| **Testing Table** | *Not included* | *A Testing Table was attempted but incorrectly completed.* | *A full Testing Table with reasonable and unreasonable data with some issues.* | *A full Testing Table with reasonable and unreasonable data.* |  |

PART A RESULT /30

**The Project B: Home DVD Library**

You are creating an application that allows you to enter your DVD library into a database with a maximum of 10 titles. The application allows you to enter your DVD titles into genres, rate them out of 5 stars, sort and search them.

**Functional Requirements**

The program needs to keep only ONE order for the user only. The program does not have to connect to a phone number and be sent through a phone network. Remember it is only a prototype. It does not transfer funds as the customer will pay on pick up.

The application is required to enter the following data Input:

* DVD Title
* Rating (1-5 stars)
* Genre (Drama, Action, Horror)

Output must include:

* Search results
* Sorted results by genre and rating
* DVD Titles must be stored to an Array

**Non-Functional requirements**

**Constraints**

You have to use the Visual Basic programming language during class time only on the following dates (Total of 6 hours):

* Wednesday 1 March (double)
* Thursday 2 March
* Friday 3 March
* Wednesday 8 March (double)
* Friday 10 march
* Wednesday 15 March (double)
* Friday 17 March DEADLINE

**Scope**

The application will only hold the DVD data in the array which must be stored in an object. The sort and search results must be displayed in the form. Only the DVD titles and ratings are required, no other identifying information (actors, directors etc.) is necessary.

**Design Layout**

My DVD Library

Search by Genre 

Search by Rating

Search by Title

Enter a DVD

DVD Title

DVD Rating

Genre

Search

Results

Enter

**Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Name** | **Data Type** | **Size** | **Description** |
| DVDTitle | String | 20 | The name of the movie |
| Rating | string | 1 | Displayed as “\*” out of five  |
| Genre | string | 15 | The type of movie |
| Index | Integer | 2 | Array Counter when sorting |
| FoundTitle | String | 20 | The DVD of the searched Title |
| Found | Boolean | TRUE/FALSE | Used to identify a found title |
| SearchTitle | String | 20 | The title that is being searched for. |
| row | Integer | 2 | Used for searching rows |
| col | Integer | 2 | Used for searching columns |

|  |  |  |  |
| --- | --- | --- | --- |
| **Array Name** | **Data Type** | **Size** | **Description** |
| DVDArray | String | 10 | This array hold DVD titles, Genres and Ratings  |

**Assessment Criteria: PART A**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Performance Descriptor** | **Not Included(0)** | **Low(1)** | **Medium(2)** | **High (3)** |  **Comments**  |
| **Variable & Object Names include appropriate type & purpose.**  | *None include appropriate type & purpose.* | *Few include appropriate type & purpose.* | *Some include appropriate type & purpose.* | *All include appropriate type & purpose.* |  |
| **Appropriate Control structures have been employed & correctly implemented.** | *Not a correct control structure for this task and doesn’t work.* | *Not a correct control structure for this task but has created a clumsy solution.* | *Correct control structure used but not well executed* | *All correct* |  |
| **Calculations** | *Not correct* | *Major issues with calculations* | *Minor issues with calculations* | *All correct* |  |
| **Clear Output** | *Not correct and unclear* | *Major issue with output* | *Minor issue with output* | *All output is clear and correct* |  |
| **Internal Documentation** | *Not included* | *Poor attempt at describing code.* | *Moderate attempt at describing lines of code.* | *Thoroughly documents all lines of codes* |  |
| **Interface Design** | *Very Poor* | *Major issues with Design of forms* | *Minor issues with design of forms* | *Excellent form Design* |  |
| **Algorithm** | *Not included* | *Included but there are many issue with the pseudocode or flowchart components and solves the problem* | *Mostly uses pseudocode or flowchart components and solves the problem* | *Correctly uses pseudocode or flowchart components and solves the problem* |  |
| **All functional and non-functional requirements are met** | *No requirements presented*  | *Few requirements presented* | *Most requirements presented* | *All requirements presented* |  |
| **Layout Design followed and improved upon** | *Not followed* | *Followed but not improved upon.* | *Followed with minor improvements* | *Followed with extensive improvements.* |  |
| **Testing Table** | *Not included* | *A Testing Table was attempted but incorrectly completed.* | *A full Testing Table with reasonable and unreasonable data with some issues.* | *A full Testing Table with reasonable and unreasonable data.* |  |

PART A RESULT /30

**The Project C: Crunch Car Hire**

Crunch Car Hire is a local business with a fleet of cars for hire in Melbourne. Mr Carl Crunch wants to streamline the booking system with a mobile app. Prospective customers can email or phone the Crunch staff who can use the application to book the car for them. The staff will use mobile devices so they can answer the queries at any stage during the day at any location on the premises.

**Functional Requirements**

The application is required to enter the following data Input:

* customer’s name
* Car type (wagon, SUV, sedan, ute, coupe, limousine)
* Number of Days (1 – 7)
* Insurance option

Calculated Output must include:

* All the entered details must be displayed as well as the calculated cost.
* Car Types must be stored as an Array.
* The cost for each number of days (1 – 7) must be stored in a file and read into a ComboBox or ListBox to the user to select.
* Today’s date must be collected from the system clock and the Due Date for return calculated and displayed.
* Buttons to CLEAR all input objects and an EXIT button must be included.

Embedded Data:

Insurance

* $20 per day is selected
* $0 per say if not selected

*Calculation Sample:*

*If an SUV was hired for 3 days with insurance the calculation would be:*

*2.0 (SUV) \* $150 (3 days) + $20 (insurance)*

CarType Array:

* wagon, 2.0
* SUV, 2.0
* sedan, 1.0
* ute, 1.5
* coupe, 3.0
* limousine 3.5

Cost per Number of Days text file:

1. $50
2. 100
3. 150
4. 175
5. 200
6. 225
7. 250

**Non-Functional Requirements**

As a prototype your application can use the dimensions for any laptop.

Within these dimensions the interface design must be intuitive for the user and follow the conventions of all mobile apps.

Font and colour choices must evoke a professional feel including the following Web colours “Light Slate Grey” and “Crimson”.

**Constraints**

You have to use the Visual Basic programming language during class time only on the following dates (Total of 6 hours):

* Wednesday 1 March (double)
* Thursday 2 March
* Friday 3 March
* Wednesday 8 March (double)
* Friday 10 march
* Wednesday 15 March (double)
* Friday 17 March DEADLINE

**Scope**

Your application is only a prototype as does not need to save any data. However it must display all calculated output to the screen. Use your knowledge of VB objects to solve you interface design problem in a creative way.

**Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Data Type** | **Size** | **Description**  |
| CustomerName | String | 25 | First and Last Name of customer |
| CarType | Floating Point | 3 | Car Type Rate |
| DateToday | Date | 10 | Today’s date from the system |
| DueDate | Date | 10 | The date the car is due |
| NoDays | Integer | 1 | Number of days of Hire |
| Price | Currency | 3 | Price set from number of days hire |
| InsuranceFee | Integer | 2 | Cost of Insurance |
| TotalCost | Floating Point | 5 | The Total cost of the booking |
| CarIndex | Integer | 1 | Counts the elements in the Car Array |

|  |  |  |  |
| --- | --- | --- | --- |
| **Constants** | **Data Type** | **Value** | **Description** |
| Insurance (Yes) | Integer | 30 | The Insurance Rate |
| Insurance (No) | Integer | 0 | No Insurance |

|  |  |  |  |
| --- | --- | --- | --- |
| **ArrayName** | **Data Type** | **Values** | **Description** |
| CarCost | Floating Point | 2.0, 2.0, 1.0, 1.5, 3.0, 3.5 | The Insurance Rate |

|  |  |  |  |
| --- | --- | --- | --- |
| **Files** | **File Type** | **Data** | **Description** |
| PriceDays | Text | 1. $50
2. 100
3. 150
4. 175
5. 200
6. 225
7. 250
 | The Insurance Rate |

**PSEUDOCODE**

On Form Load (When the application starts up)

Start

 Declare variables

 Find Text File (PriceDays)

 Open PriceDays.txt

 Repeat Until EOF

 Read NoDays value

 Add NoDays value to object

 Read Price value

 Add Price value to object

 End Loop

 Close File

End

On Car Type Object selection

Start

 Declare CarCost Array with values

Declare CarIndex variable

Selected Index value from Selction object

CarType = Index Value

Display CarType as Currency in label

End

On Activating Calculation Button

Start

 Input CustomerName

 IF Insurance = YES THEN

 InsuranceFee = 30

 ELSE

 InsuranceFee = 0

END IF

 Read Selected NoDays to display Price

 TotalCost = (CarType \* Price) + InsuranceFee

 Display CustomerName

 Display CarType

 Display Price

 Display NoDays

 Display InsuranceFee

 Display TotalCost

End

**Assessment Deliverables**

1. Create the fully working prototype solution for Crunch Car Hire
2. Complete the TESTING TABLE below for your solution to check ALL possible outputs.

|  |  |  |  |
| --- | --- | --- | --- |
| Item Tested | Test Data | Expected Result | Actual Result |
|  |  |  |  |
|  |  |  |  |

1. Internal documentation that describes the function of each line of code.

**Assessment Criteria: PART C**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Performance Descriptor** | **Not Included(0)** | **Low(1)** | **Medium(2)** | **High (3)** |  **Comments**  |
| **Variable & Object Names include appropriate type & purpose.**  | *None include appropriate type & purpose.* | *Few include appropriate type & purpose.* | *Some include appropriate type & purpose.* | *All include appropriate type & purpose.* |  |
| **Appropriate Control structures have been employed & correctly implemented.** | *Not a correct control structure for this task and doesn’t work.* | *Not a correct control structure for this task but has created a clumsy solution.* | *Correct control structure used but not well executed* | *All correct* |  |
| **Calculations** | *Not correct* | *Major issues with calculations* | *Minor issues with calculations* | *All correct* |  |
| **Clear Output** | *Not correct and unclear* | *Major issue with output* | *Minor issue with output* | *All output is clear and correct* |  |
| **Internal Documentation** | *Not included* | *Poor attempt at describing code.* | *Moderate attempt at describing lines of code.* | *Thoroughly documents all lines of codes* |  |
| **Interface Design** | *Very**Poor* | *Major issues with Design of forms* | *Minor issues with design of forms* | *Excellent form Design* |  |
| **File Data Management** | *Not included* | *An attempt to import data from a text file which did not operate in the selection object.* | *Text file data was correctly formatted but there were minor issues with importing to a selection object.* | *Text file data was correctly formatted and imported to a selection object.* |  |
| **All functional and non-functional requirements are met** | *No requirements presented*  | *Few requirements presented* | *Most requirements presented* | *All requirements presented* |  |
| **Local and Global operations and Control Structures**  | *Not attempted* | *Local and Global operations and Control structures attempted.* | *Local and Global operations and Control structures integrated with minor issues.* | *Local and Global operations and Control structures correctly integrated* |  |
| **Testing Table** | *Not included* | *A Testing Table was attempted but incorrectly completed.* | *A full Testing Table with reasonable and unreasonable data with some issues.* | *A full Testing Table with reasonable and unreasonable data.* |  |
| **Problem Solving** | ***0 – 2****The student has not solved the problem using any appropriate solution components (control structures, objects, data management techniques). Marks for effort.* | ***3 – 4****The student has attempted to solved the problem using some appropriate solution components (control structures, objects, data management techniques) for an incomplete solution* | ***5 – 7****The student has solved the problem using some appropriate solution components (control structures, objects, data management techniques) for an adequate solution* | ***8 – 10****The student has solved the problem using the most appropriate solution components (control structures, objects, data management techniques) for an efficient solution.*  |  |

PART A RESULT /40