|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| US18740 | People credited with this unit standard are able to: plan a simple computer program to meet the specifications of a set brief; create draft code; and evaluate the program against the brief for fitness for purpose. | version 6 | Level | **2** | **Credits** | **3** |

**Assessment Task: 2022**

**Brief:**

Making Dough

Mr Knead is your local baker. He recently bought out a number of smaller bakeries scattered around the city. To determine each store’s success, he asks the manager to submit the store’s daily sales. At the end of each week Mr Knead creates a summary showing the figures for each store.

This is a rather tedious task and Mr Knead has employed you to write a computer program to automate the process. Your program needs to…

* Record the Store’s name
* Record the daily sales for the week
* Find the lowest daily sales amount
* Find the highest daily sales amount
* Find the ‘average’ (lowest + highest)/2
* Report the GST component of sales . Note GST is currently 15%

Mr Knead is hoping to add more stores to his rising empire. Your program should loop until the user enters ‘xxx’ or ‘XXX’ for the store name. Once that code has been entered, the program should

Print the stores in alphabetical order with their weekly sales

Hints:

Plan your work!

You should add to the test data as appropriate. Testing for both edge values and unexpected values.

* Create a list to hold the summary data for all the stores
* Create a ‘temporary’ list to hold the data for each store. This can be appended to the overall summary list
* Use [start:end] notation to access part of a list ie print(a[2:5]) *# prints [3, 4, 5]*
* The methods below might be useful
  + round(item,decimal\_places)
  + min(list\_or\_part\_of\_list)
  + max(list\_or\_part\_of\_list)
  + list.sort()
* Use functions to check that inputs are valid and to avoid repeating code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * **Elements and performance criteria** | | | | |
| **Element 1**  **Performance Criteria** | Plan a simple computer program to meet the specifications of a set brief. | | Student | Teacher |
| Date | Date |
| 1.1 (a) | A plan is developed to meet the specifications of the given brief.  Must show:  Milestones/timeline (x3), resources (at least2 ), stakeholder consultations (at least 3), testing procedures | |  |  |
| 1.2 (a) | The plan outlines a logical design or model to be used as a basis to code the program:  Including suitable flow chart  a. Clearly outline the purpose of the program  b. Specify the computer language to be used | |  |  |
| 1.2 (b) | c. Clearly explain the input and output requirements | |  |  |
| 1.2 (c) | d. Clearly outline the proposed testing procedures | |  |  |
| **Element 2**  **Performance Criteria** | | Create draft code. | Student | Teacher |
| Date | Date |
| 2.1 (a) | | The code is created in **accordance with the plan.**  *May include but is not limited to – sequence, selection, repetition, conditional execution, variables, labels, procedures and functions.*  *A minimum of four structural elements are required.* |  |  |
| 2.1 (b) | | Clear evidence of testing procedure followed.  (Print-Screens are required as evidence) |  |  |
| 2.2 | | Internal names are applied to all user-defined elements of the program code to facilitate understanding of the program. |  |  |
| 2.3 (a) | | The draft code is internally documented according to the requirements of the brief.  Each section of code must contain sufficient comments to explain its purpose. |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Elements and performance criteria** | | | |
| **Element 3**  **Performance Criteria** | Evaluate the program against the brief for fitness for purpose. | Student | Teacher |
| Date | Date |
| 3.1 (a) | The program is tested for compliance with the brief and modified as required.  (Print-Screens will count as evidence) |  |  |
| 3.2 (b) | The evaluation verifies that the operation of the program realises the specifications set out in the brief. |  |  |

**Student Notes:**

1. It is your responsibility to ensure that you submit all required evidence for this Unit Standard.
2. You need to have your name as a header on all documents and a footer with page numbering
3. As you work through this assessment fill in date and tick off work that you have completed in the student column.
4. If you are sure that you have clearly met the Performance Criteria have your teacher provisionally mark off the Teacher column. Do this at each stage – not as a big chunk at the end. Make sure you are working correctly as you go.
5. The teachers mark is a general indication only. The final mark will be decided after formal submission.
6. Final code (suitably labelled ) must be available to the assessor in your H drive then stuwriteonly

**This assignment is issued on, Wednesday 6th April 2022**

**Due Date for submission is Tuesday 24th May by 8.30am**

**There are no late submissions**

On Monday the 13th of May you will be given a set of test data to enter The results of this will be copied to the last page of your report as **Teacher Testing**

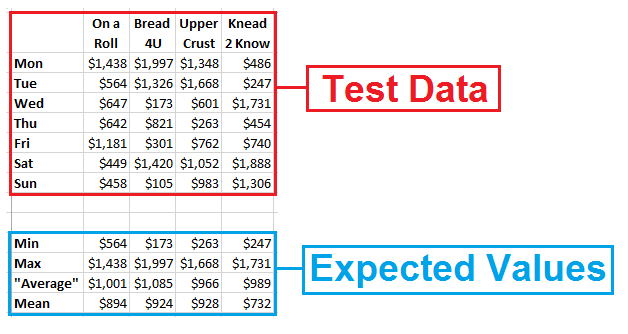
You will be given 3 class periods for 4 weeks to work on this project

There needs to be at least **three** stakeholder consultations (The stakeholder in this case will be your teacher and another student who will be expected to provide written feedback about your program which you will include in your write up) over the period of your assignment.

Consult with your teacher to schedule in stakeholder consultations in your study periods if possible, or during lunch.

Remember 30 students within a 50 min period only provides 100 seconds of time per student in class (not accounting for any travel time between workstations)

# Test Data



|  |  |  |  |
| --- | --- | --- | --- |
| **Name** |  | **US 18740** | Achieved / Not Achieved |
| **Date** |  | **Further evidence required** |  |
| **Conditions** | Open book assessment over 6 periods | **Authenticity** | All work is observed to be students own. Examination conditions apply. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EVIDENCE** | | **JUDGEMENT** | | | |
| **ELEMENT 1 – Plan a computer program.** | | | | | |
| 1.1 A plan is developed to meet the specifications of the given brief.  Range milestones, resources, stakeholder consultations, testing procedures. | A plan is produced.  The plan may be presented as a flow chart. Input and output requirements must also be shown.  Milestones & Resources to be used must be listed.  Stakeholder consultations & testing procedures are listed. | | Plan must demonstrate understanding of the brief and offer a solution to the problem.  Milestones & resources used listed.  Testing procedures described. | 1.1 Theory - Written | 🞏  🞏  🞏 |
| 1.2 The plan outlines a logical design or model to be used as a basis to code the program.  Range the design or model includes but is not limited to – purpose, structure and specifications of the program. Specifications may include but are not limited to – computer language to be used, input and output requirements, outline of the proposed testing procedures. | Purpose of the program is clearly stated.  Specifications & criteria of the program are listed.  The intended audience is clearly stated.  Computer language used is stated.  Proposed testing procedures is listed | | Local club is identified and the reason the club needs the program is identified.  *Eg: My program is for the local “Friendly Services” club. They require a simple program to calculate the bill for services.*  The type of data that is to be entered is listed and format expected is shown.  *Eg: Elements – Name Type of service ie Taxi, Hair dressing, drinks and cost of each*  The audience of the program is stated and they are described.  *Eg: This club provides services to 25 to 90 year old*  The computer language used is stated.  *Eg: Javascript, C++, Python*  Testing procedures to be used at each step are specified.  *Eg: Procedures will identify what is to be tested and include testing at each step with a range of data types, blank or null values, a range of values, and testing of form controls.* | 1.2 Theory - Written | 🞏  🞏  🞏  🞏  🞏 |
| **ELEMENT 2 – Create draft code.**  Range may include but is not limited to – input, selection, repetition, conditional execution, variables, labels, procedures and functions, output | | | | | |
| 2.1 The code is created in accordance with the plan. | Draft code is created according to plan. | Any code is acceptable as long as it is in accordance with the plan. | | 2.1 Practical- Program | 🞏 |
| 2.2 Internal names are applied to all user-defined elements of the program code to facilitate understanding of the program. | Labels, variables and procedures are user-defined.( list of type and nature) | Names chosen will be appropriate and relevant to the program. | | 2.2 Practical- Program | 🞏 |
| 2.3 The draft code is internally documented according to the requirements of the brief.  Range each section of code must contain sufficient comments to explain its purpose. | Relevant non-executable comments that explain the purpose of the code is present. | Each section must be commented and clearly explained. | | 2.3 Practical- Program | 🞏 |
| **ELEMENT 3 – Evaluate the program against the brief for fitness for purpose.** | | | | | |
| 3.1 The program is tested for compliance with the brief and modified as required. | Modifications are made to ensure it complies with the brief. | Evaluation must comply with the brief. | | 3.1 Practical- Program | 🞏 |
| 3.2 The evaluation verifies that the operation of the program realises the specifications set out in the brief. | The program is checked for errors, and modifications are made where required. | Checking is completed in accordance with the testing procedures listed. | | 3.2 Practical- Program | 🞏 |