## Computer Program Design Plan for

## “Baker Program with GUI”

|  |  |  |  |
| --- | --- | --- | --- |
| * + 1. **Program function** | | | |
| *Indicate the function/purpose of the computer program (i.e. what does it need to be able to do and why).* | | | |
| The purpose of this program is to help a local baker start up to calculate the costs of his labour. So he can record his customer’s orders in an order book or logbook. He currently has three types of bakery products, for which he charges different prices (quantity of products is already pre-described)   * Cookies (A Pack of 12 Assorted Flavours) for **$16.20** a pack * Cupcakes(A Pack of 6 Assorted Flavours) for **$21** a pack * Cake(Chocolate Flavour Only)for **$59.50**   There is however a flat travel fee of $10 for delivery within a 20km radius from his bakery. The owner himself will check whether the actual order made is within 20km radius. The program needs to be able to handle multiple orders, and needs to calculate and display the following:   * The Quantity of each baked product * The Total Cost for each baked product * The Overall Total Cost to charge customers   Important details to appear in order book: **(Extra Features)**   * **Date Format (Day of order)** * **Max & Min Order Type Boundaries (Cookies – 10, Cupcakes – 10, Cakes - 5)** * Customer Details (First & Last Name) * Order Details (Quantity of Baked Products & Cost)   This will all be coded with a GUI. For better display of information. | | | |
| * + 1. **Target language** | | | |
| *Indicate the target programming language (e.g. Python 3 incl. version) for the computer program.* | | | |
| *Python 3.5.1 Whilst Importing other modules, such as baker\_calculator, tkinter and ttk, messagebox from tkinter, datetime.* | | | |
| * + 1. **Data dictionary** | | | |
| *Indicate the data items, data types, scope (local/global) and the variables/constants (incl. functions/modules, lists, derived values). An example is given under the column headings.*  *Remember:*   * *you need to create 2 or more user-defined functions/modules (include: comments to explain the purpose of each user-defined function/modules)* * *any modules you create needs to be correctly named, and saved as py file* | | | |
| **Data item**  ***Example***:  *User’s age (i.e. get\_userage function is used to ask for user’s age and store it in a local variable called user\_age for use when the function is called)* | **Data Type**  *Number - integer* | **Variable Scope (local / global)**  *local* | **Variable/Constant (incl. functions/modules, lists, derived values)**  *user\_age* |
| Global variables that contain the **cookies**, **cupcakes** and **cake** **prices**, it is a constant and thus does not change throughout the program. | **Integer** | **Global** | **COOKIES\_PRICE** |
| **CUPCAKES\_PRICE** |
| **CAKE\_PRICE** |
| A global variable that contains the travel fee. | **Integer** | **Global** | **TRAVEL\_FEE** |
| Global variables that contain separate amounts of maximum and minimum amount of orders. It is a constant and will not be changed throughout the program. | **Integer** | **Global** | **COOKIES\_MAX** |
| **CUPCAKES\_MAX** |
| **CAKE\_MAX** |
| A variable that will hold the current date and time of the day that the program is run, which will then be used to write to file to measure the date. | **String** | **Global** | **date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| * + 1. **Algorithmic structure (Procedural / Functions & Methods / Objects & Classes)** | | | |
| *Indicate the algorithm for the program using a suitable algorithm tool(s). For more information, see notes.*  ***Remember:*** *You may use a combination of pseudocode and decomposition diagram(s).* | | | |
| *Hyperlink all versions of pseudocode (comment errors found and fixed for each version)*  *(Terminal Reliant Program)*  [*baker\_program\_pcode\_versions\pcode\_1.txt*](file:///\\stbernards.school.nz\Students\Private_Folders\EBugas1\01%20DTC\US%20-%2018741\Assessment%2018741\US18741%20Assessment%202019\18741v6%20Programming%20Assessment%20-%20Ellan%20Bugas\baker_program_pcode_versions\pcode_1.txt)  [*baker\_program\_pcode\_versions\pcode\_2.txt*](file:///\\stbernards.school.nz\Students\Private_Folders\EBugas1\01%20DTC\US%20-%2018741\Assessment%2018741\US18741%20Assessment%202019\18741v6%20Programming%20Assessment%20-%20Ellan%20Bugas\baker_program_pcode_versions\pcode_2.txt)  [*baker\_program\_pcode\_versions\pcode\_3.txt*](file:///\\stbernards.school.nz\Students\Private_Folders\EBugas1\01%20DTC\US%20-%2018741\Assessment%2018741\US18741%20Assessment%202019\18741v6%20Programming%20Assessment%20-%20Ellan%20Bugas\baker_program_pcode_versions\pcode_3.txt)  *(Gui Reliant Program)*  *Hyperlink all versions of the program code (comment errors found and fixed for each version)*  *(Terminal Reliant Program)*  [*baker\_program\_versions\baker\_version1.py*](file:///\\stbernards.school.nz\Students\Private_Folders\EBugas1\01%20DTC\US%20-%2018741\Assessment%2018741\US18741%20Assessment%202019\18741v6%20Programming%20Assessment%20-%20Ellan%20Bugas\baker_program_versions\baker_version1.py)  [*baker\_program\_versions\baker\_version2.py*](file:///\\stbernards.school.nz\Students\Private_Folders\EBugas1\01%20DTC\US%20-%2018741\Assessment%2018741\US18741%20Assessment%202019\18741v6%20Programming%20Assessment%20-%20Ellan%20Bugas\baker_program_versions\baker_version2.py)  [*baker\_program\_versions\baker\_version3.py*](file:///\\stbernards.school.nz\Students\Private_Folders\EBugas1\01%20DTC\US%20-%2018741\Assessment%2018741\US18741%20Assessment%202019\18741v6%20Programming%20Assessment%20-%20Ellan%20Bugas\baker_program_versions\baker_version3.py)  *Final -* [*baker\_program\_versions\baker\_version4.py*](file:///\\stbernards.school.nz\Students\Private_Folders\EBugas1\01%20DTC\US%20-%2018741\Assessment%2018741\US18741%20Assessment%202019\18741v6%20Programming%20Assessment%20-%20Ellan%20Bugas\baker_program_versions\baker_version4.py)  *(Gui Reliant Program)* | | | |
| **Algorithmic structure** | | | |
| *decomposition diagram(s) – shows the main routine of program (final version of program)* | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * + 1. **Testing procedures** | | | | |
| *Indicate test cases for testing the planned program.* | | | | |
| |  |  |  |  | | --- | --- | --- | --- | | **Test:** | **Expected Result:**  **Refers to Algorithm (incl. pseudocode)**  (used for Desk checking) | **Actual Result:**  **Refers to program**  (see screenshot section below) | **Result: Pass / Fail?**  **Incl. comments relating to “find and fix” errors** | | **EXPECTED / NORMAL INPUTs** *(at least 4 test cases)* | | | | |  |  | Figure 1 |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | **BOUNDARY / EXTREME INPUTs** *(at least 6 test cases e.g. 3 lower boundary and 3 upper boundary for each input)* | | | | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | **EXCEPTIONAL / INVALID INPUTs** *(at least 4 test cases)* | | | | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  |   **Screenshots of Testing:**  ***Remember*:**   * *Clearly label each screenshot (e.g. figure 1); and* * *State which test case each screenshot relates to in your testing table*   **Screenshots Evidence for Expected Inputs**  **Figure 1**  **Screenshots Evidence for Boundary Inputs**  **Screenshots Evidence for Invalid Inputs** | | | | |
| 1. **Project Plan** | | | | |
| **Progress Plan (AS 91637)**  *You may use this “Progress plan” table to help manage your time, resources, keep discussion notes (client) and document steps you took to troubleshoot issues/problems you encountered.*  *You can hyperlink files to show evidence of meetings with client, troubleshooting issues with pseudocode (incl. versions) and program(incl. versions)* | | | | |
| **Resources Required (incl. software and hardware):** | | | | |
| ***Date and Time***  ***(incl. period, minutes):*** | ***What I did?*** | ***State any issues/problems:*** | ***What I did to solve them?*** | ***Discussion notes***  ***(Client meetings)***  **Hyperlink meeting notes below** |
| **Friday 14 June**  **(pd 1, ? mins)** | **Created folders and files (client meeting notes)**  **Section A - Brief**  **Section B – Program Language and version** | **None** | **None** |  |
| **Monday 17 June**  **(pd 3, 60mins)** | **Finished First GUI Design. Implementing the design into code.** | **N/A** | **N/A** |  |
| **Tuesday 18 June**  **(pd 5, 60mins)** | **Finished Implementing Design into Code, working on next second design of the GUI.** | **N/A** | **N/A** |  |
| **Thursday 20 June**  **(pd 1, 55mins)** | **Not here** |  |  |  |
| **Friday 21 June**  **(pd 2, 60mins)** | **Not here** |  |  |  |
| **Monday 24 June**  **(pd 2, 60mins)** | **Creating Second GUI Design. AS the first was not finished and missing buttons. Adding new features as well..** | **Stuff wasn’t aligning…** | **N/A** |  |
| **Tuesday 25 June**  **(pd 4, 60mins)** | **Not here** |  |  |  |
| **Wednesday 26 June**  **(pd 5, 60mins)** | **Code is finally completed, fixing pcode.** | **N/A** | **N/A** |  |
| **Friday 28 June**  **(pd 1, 60mins)**  **– refer checklist (parts to submit – check and amend)** | **Pcode and more documenting needed to be done.** | **N/A** | **N/A** |  |
| **Monday 1 July**  **(pd 3, 60mins)** | **Data Dictionary and Testing** | **N/A** | **N/A** |  |
| **Tuesday 2 July**  **(pd 5, 60mins)** |  |  |  |  |
| **Thursday 4 July**  **(pd 1, 60mins)**  **(submit all printouts and digital files)**  **DUE TODAY** |  |  |  |  |

|  |
| --- |
| 1. **End-User Guide** |
| *The End-User Guide must include the following:*   1. *The Purpose or description of the program* 2. *Instructions on how to access the program* 3. *Instructions on how to use the program* |

|  |
| --- |
| 1. **Evaluate** |
| *Evaluate the following (provide comments):*   1. *Does the program function as it should?* 2. *Does the program solve the problem and address all the requirements of the brief?* 3. *Has the program and the end-user guide been checked for readability, legibility, and presentation?* 4. *Is the program accurate (and error free)?* 5. *Does the program comply with the plan?* 6. *Is the required documentation complete?* 7. *Have modifications been noted and explained?* 8. *Describe improvements for the next version of your program i.e. a) which parts/features of your program requires improvement and why?*   *b) Which parts/features can be easily adapted and how?* |