# BIT103 PROGRAMMING FUNDAMENTALS

# Assessment 2— Problem Solving Assignment

|  |  |  |  |
| --- | --- | --- | --- |
| **Due date:** | 11:55pm AEST Sunday, November 5th | **Individual/Group:** | Individual |
| **Weighting:** | 20% | **Total Marks:** | 100 marks |
| **Length:** | N/A | | |

**Task Summary**

This is an individual assessment that requires students to write correct Python programs to implement specific tasks. Given a case study, students need to write Python code to create data objects using appropriate data types and complete required tasks using appropriate control statements. This assessment contributes 20% towards the final results.

**Assessment Description**

A Python program that stores and processes student result records is in its early stage of development. The student records are stored in a text file called **class.txt**. In this file, student’s names, student IDs and their marks are saved. Each student occupies one line in the file and each field is separated by tabs.

A menu is used to list the processing tasks that can be applied on the file. Some of the menu items have been coded; others have a stub (dummy) code provided. The menu program is provided in a Python file called **menu.py**.

**Submission Instruction**

* You are expected to submit a report written in MS Word. Your report needs to answer all the questions in Assessment Tasks - Section A. Your report should be named using your student ID, for example, ‘**xxxxxxxxxx\_A1report.doc**’.
* The report needs a title page with the unit code, unit name, assessment title, student ID, student name, and date. Correct report format needs to be used with appropriate reference if necessary.
* You need to submit the Python file containing the complete program that all the menu items can accomplish the corresponding processing tasks as described in Assessment Tasks – Section B. You need to rename the file to add your student ID, thus a sample file name can be ‘**xxxxxxxxx\_menu.py**’.
* You also need to include the **class.txt** file in the submission, but no need to make any change.
* You can compress all the files into a single one (zipped) and upload to Moodle before the due time.

**Assessment Tasks**

**Section A: Questions for Report (45 marks)**

1. Debug and run **menu.py** file from the assessment specification package. Try each option in the menu and write down the outputs respectively. **(8 marks)**
2. Draw a structure diagram to show the design (structure) of this program. **(12 marks)**
3. Line 3 in main() function is infile = open(infileName, 'r'). What type of statement it is? Explain each component in the statement respectively using correct programming terminologies. **(4 marks)**
4. List at least three advantages of having many small functions in a program rather than put everything in the main() function. **(3 marks)**
5. Write pseudocode for the missing functions in option 3 to 8 respectively. **(3\* 6 = 18 marks)**

**Section B: Coding in menu.py file (55 marks)**

Modify the program according to the following instructions.

1. Menu item 2: “Display the student list with student IDs” does not return the correct output. Find the corresponding function, identify the problems and fix the code. **(4 marks)**
2. Menu item 3: “Display the student name given a student ID” insert the correct code to complete the corresponding function in the program. **(4 marks)**
3. Menu item 4: “Display the student’s mark given a student ID” insert the correct code to complete the corresponding function in the program. **(4 marks)**
4. Menu item 5: “Display the maximum mark” insert the correct code to complete the corresponding function in the program. **(6 marks)**
5. Menu item 6: “Display the minimum mark and corresponding student ID” insert the correct code to complete the corresponding function in the program. **(6 marks)**
6. Menu item 7: “Display all the students who has a pass mark” insert the correct code to complete the corresponding function in the program. **(5 marks)**
7. Menu item 8: “Display student list in grade groups” insert the correct code to complete the corresponding function in the program. The grade groups are defined as group HD: 80-100; D: 70-79; C: 60-69; P: 50-59; and N: 0-49. **(8 marks)**
8. Modify the main() function to insert the appropriate code into each menu option to make the entire program work correctly. **(8 marks)**
9. Provide appropriate comments for the program. **(10 marks)**

**Marking Guide**

**Section A: Questions for Report (45 marks)**

1. Missing output for any option (-1 mark each)
2. Structure diagram
   1. Missing any function (-1 mark each)
   2. Wrong connection (-0.5 mark each)
   3. Wrong notation (-0.5 mark each)
3. Statement explanation
   1. Wrong statement (-1 mark)
   2. Wrong component explanation (-1 mark each)
4. Missing any advantage (-1 mark each)
5. Wrong pseudocode for any function (-3 mark each)

**Section B: Coding in the Python file (55 marks)**

The following marking criteria are used for all the questions in this section.

1. Wrong statement (-1 mark each)
2. Syntax error (-1 mark each)
3. Wrong output (-1 mark each)
4. Wrong argument or parameter (-0.5 mark each)
5. Wrong placement of the statement (-1 mark each)
6. Missing key comment (-1 mark each), -10 marks if no comments in the new functions at all