



UNIVERSITI TUNKU ABDUL RAHMAN

FHCT1024 PROGRAMMING CONCEPTS AND DESIGN

FOUNDATION IN SCIENCE (Session 202210)

GROUP ASSIGNMENT GUIDELINES

ASSESSMENT: The assignment constitutes **20%** of the final grade

ASSIGNMENT THEME:
"University Assessment Grading System"

Title	Page
1.0 Objectives	2
2.0 Assignment Rules	3
3.0 Requirements / Output	4
4.0 Problem / Question	5
5.0 Grading	6
6.0 Screenshots	7 - 10

This assignment will assess the following Course Learning Outcomes (CLO) and Domain:

CLO	CLO Description	Domain
CLO 3	Demonstrate ways in solving computing problems using various control structures and text files management.	Cognitive

1.0 Objectives

This assignment is designed to enable you to put into practice the knowledge and concepts you have acquired over the trimester on problem solving and Python Program development to solve a computing problem effectively.

A problem will be provided to you where you will need to employ the program development processes (Six (6) Steps of Program Development) to assist you to build a viable system. These six (6) steps include *defining the problem*, *developing an algorithm*, *testing the algorithm*, *coding the algorithm*, *running the program*, and *documenting & maintaining the program*. [For more details, refer to Lecture Topic 1.]

You are expected to fully cooperate with your team members to take a detailed study of the problem (domain knowledge), decompose a complex problem into smaller, manageable parts so that the entire team can work hand in hand to build efficient and effective modules and functions that can be integrated easily to form an integral solution.

You are expected to inculcate the needed level of problem-solving, coding skills as well as to acquire all related soft skills needed in solving a computing problem in a team at the end of this assignment.

2.0 Assignment Rules

1. **Submission Date:** 27th December 2022, Tuesday (Week 11) before 5pm.
2. **Group size:** 4-5 persons in a group.
3. **Submission:** Your submission, in the form of a zipped folder must include:
 - ONE (1) report (.docx), and
 - ONE (1) or more source code and text files (.py & .txt)**n/b: Your submission will be via Google Form for CFS-KPR's students and via WBLE for CFS-SL's students. Submission links will be made available to you on WBLE.**
4. **Penalty:** No marks (0%) will be awarded to everyone in the group if **plagiarism** is found, **and** 10% of the entire assignment marks will be deducted each day for **late submission**.
5. **Responsibilities:**
 - You are **to form** your own assignment group and choosing your own group leader.
 - You are to establish your own communication line using any of the collaboration tools among your teammates.
 - Each of you **MUST** contribute sufficiently to the completion of the assignment. If you are found to be unresponsive or uncooperative or do not contribute sufficiently to the assignment, your lecturer has the **SOLE discretion** to award a different grade to you for your work/effort.

3.0 Requirements / Output

1.0 Requirements for report:

- You are expected to prepare a **NEAT** and **LEGIBLE** report.
- You are to include **ONLY** those sections listed (table of contents, flowchart, program, screen shots of the system) in the template. **NO additional section is allowed.**

2.0 Requirements for algorithm:

- You can use any flowchart-drawing applications to draw your flowcharts. **Hand drawn flowcharts are not allowed.**
- You are advised to construct well-documented algorithm(s) (flowcharts) showing all the detailed steps, in a correct manner, on how the entire system works.
- You are to use multiple pages with clearly labelled page connectors to document your solution and **NOT** to cram too many symbols into one page. You must provide the right symbols for each of the algorithmic steps you use to solve the problem. You will also need to use the correct font and size (standardised to either Arial or Times New Roman) for your labelling.

3.0 Requirements for Python Program:

- Program written must:
 - be error-free and can produce the right output.
 - be well-thought-out, with the use of the right combination of control structures and subprograms (self-defined functions).
 - be well-presented screen interfaces with the ease to navigate from one screen to another.
 - include data validation, along with, meaningful error messages to alert the user when he or she keys in invalid data.
 - be well documented (internal documentation) but not excessively.
- Your program code must be tidy and readable. You are advised to adhere to a good coding standard and style.
 - Adopt a naming convention for constants or variables (e.g. prefix of g for global variables).
 - Use tab instead of space for indentation (4 spaces for indentation).
 - Program block (number of steps in each self-defined function/control structure) should not be too large/small.
 - Use equal-width font (e.g. courier new) for the codes in the report.

4.0 Choose to enhance the system to include special techniques, features and functions that can simulate actual operations of such system.

4.0 **Problem / Question**

You are required to develop a system that can be used to calculate the results of students who have signed up for courses in UTAR. The system should, at the very minimum be able to operate as such:

1. Create, Read, Update, and Delete (CRUD) system maintenance for:
 - a) Courses (course code and description)
 - b) Assessment (types, total marks and weightage)
 - c) Students' information (student ID and name)
2. Enter students' marks for each of the assessments.
3. Compute and display the list of students' results based on course codes to the screen accordingly.
4. All students' records for students' information, courses, course-assessment types, course-students and marks are to be stored in text file(s).

You are required to have the following features in your system:

1. A minimum of **THREE (3)** courses.
2. A minimum of **TEN (10)** students for each course.

Please conduct your own research to gain additional knowledge on ways to enhance the system so that it will be reliable, scalable and maintainable. You may also want to include the following features and functions.

1. Allow authorised users to access the system by entering login ID and password.
2. Generate a template (.txt file) to enter students' marks for each of the assessments. Then, upload the text file to the system for result calculation.
3. Compute and display course performance analysis (average, standard deviation, failure rate, number of students for each grade obtained, etc.) to the screen.
4. Other reports or features you may have identified.

5.0 **Grading (100%)**

1. **Algorithm (Flowcharts) (15%)**

Flowcharts show each step of the solutions in a detailed, clear, and precise manner. Flowcharts are drawn using the correct symbols, line arrows and labels.

2. **Program Specifications / Correctness (30%)**

All parts of the program work correctly without errors. All basic specifications are met. Your program must produce all expected output accurately. Your program is written according to the steps depicted / represented in the flowcharts.

3. **Program Control and Subprogram Structures (Code Efficiency) (15%)**

The order in which program instructions are performed should be carefully controlled, and programming problems are reduced to combinations of controlled sequences, selections, and repetitions (loop) in the most efficient manner. You **MUST NOT** use a recursive function call in your program.

Programs are organised into subprograms / self-defined functions and are self-contained (independent of other parts of the program) with well-defined data (locally declared variables) and operations.

4. **Interface Design and Flexibility in Navigating between & within Functions and Error Rectification (10%)**

Clear instructions are provided on the screen for users to navigate through the functions / subprograms / menus / submenus of the system efficiently. Users are also guided with the understanding of what data and data structures to be entered to allow the system to process it into the expected output.

5. **Validation and Error Messages (10%)**

Proper validation and control are placed to check for correct data type, data range, data length / size, and completeness depending on the use of data structures in the system. Error messages on what goes wrong must be clearly displayed to alert users. Users must be directed clearly on how to go about rectifying the problem.

6. **Documentation and Coding Style (Code Appearance) (10%)**

Your Python Program is documented in a neat and tidy fashion. Proper header and meaningful comments (internal documentation) are provided accordingly in the program. Line spacing and spaces used in the statements are standardised to allow readability.

7. **Enhancement of System (10%)**

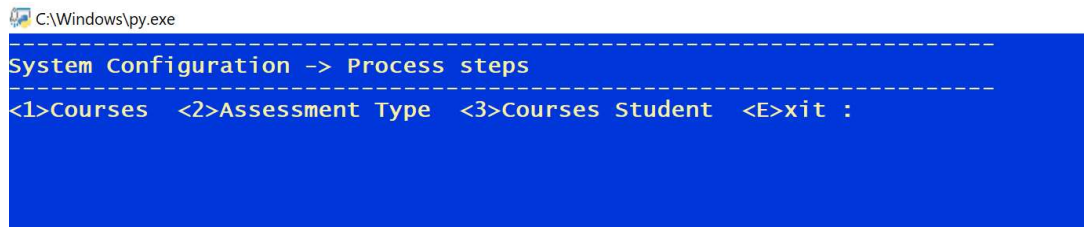
Your program is enhanced to cater for features and functions that resemble those of a live (actual working) environment. It is also structured in such a way that it is configurable or adaptable to changes in requirements (minimum changes are required to be made to the program and structure when there is a requirement change) – having little “hardcoding” of any sort.

6.0 Screenshots

Note:

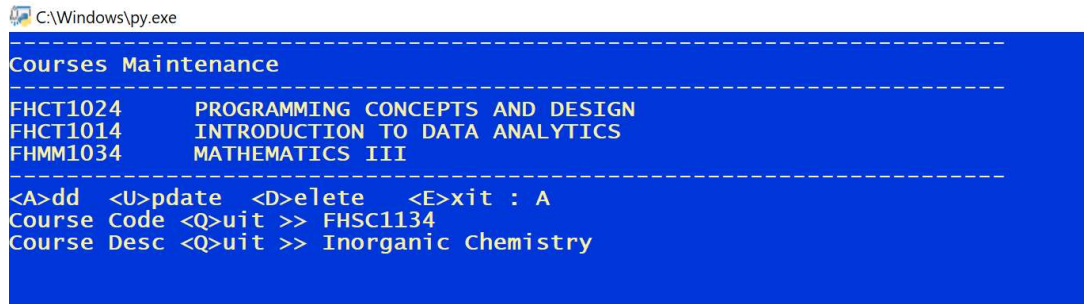
- The following screen shots show some sample interfaces of different functions or modules.
- It is not compulsory to follow the design of the following interfaces. You are encouraged to design your own interfaces.

6.1 System Maintenance Configuration



```
C:\Windows\py.exe
-----
System Configuration -> Process steps
-----
<1>Courses <2>Assessment Type <3>Courses Student <E>xit :
```

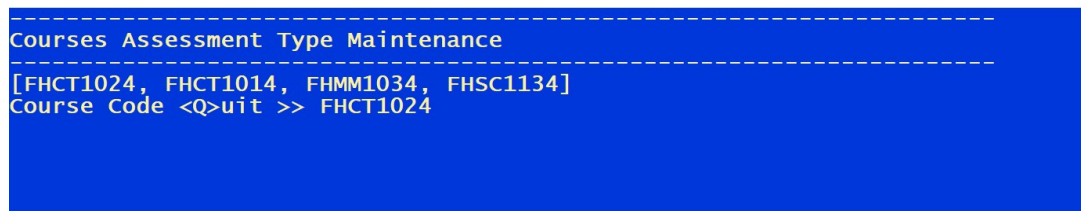
6.2 Course Maintenance



```
C:\Windows\py.exe
-----
Courses Maintenance
-----
FHCT1024    PROGRAMMING CONCEPTS AND DESIGN
FHCT1014    INTRODUCTION TO DATA ANALYTICS
FHMM1034    MATHEMATICS III
-----
<A>dd <U>pdate <D>elete <E>xit : A
Course Code <Q>uit >> FHSC1134
Course Desc <Q>uit >> Inorganic Chemistry
```

6.3 Assessment Maintenance

6.3.1 Enter Course Code



```
-----
Courses Assessment Type Maintenance
-----
[FHCT1024, FHCT1014, FHMM1034, FHSC1134]
Course Code <Q>uit >> FHCT1024
```

6.3.2 Assessment Maintenance for FHCT1024

```
C:\Windows\py.exe
-----
Courses Assessment Type Maintenance
-----
FHCT1024 (PROGRAMMING CONCEPTS AND DESIGN)---->
T1  50  0.10
T2  50  0.20
A  100  0.20
F  100  0.50
-----
<A>dd  <U>pdate  <D>elete  <E>xit :
```

6.3.3 Assessment Maintenance for FHCT1014

```
C:\Windows\py.exe
-----
Courses Assessment Type Maintenance
-----
FHCT1014 (INTRODUCTION TO DATA ANALYTIC)---->
T   50  0.20
L   20  0.10
A  100  0.20
-----
<A>dd  <U>pdate  <D>elete  <E>xit : A
(T#)est/<P#>ractical/<L#>ab/<A#>ssignment/<F>inal Exam
Assessment Type      <Q>uit >> F
Total Marks          <Q>uit >> 100
Total Marks to Final <Q>uit >> 0.5
```

6.4 Student Maintenance

6.4.1 Enter Course Code

```
C:\Windows\py.exe
-----
Courses Student Maintenance
-----
[FHCT1024, FHCT1014, FHMM1034, FHSC1134]
Course Code <Q>uit >> FHCT1024
```


6.4.2 Student Maintenance for FHCT1024

```
C:\Windows\py.exe
-----
Courses Student Maintenance
-----
FHCT1024 (PROGRAMMING CONCEPTS AND DESIGN)--->
2200012 John Anthony
2200011 Jamie Crossby
2210001 Tan Yin Jing
2113001 Chia Xi Xin
2217033 Ahmand Ali
2200013 Natasha
2200014 Daniel
2200015 Adrian Wong
2200016 Soo Ai Ling
-----
<A>dd <U>pdate <D>delete <E>xit : A
Student ID (YY####) <Q>uit >> 2200017
Student Name <Q>uit >> Desmond Tan Sok Man
```

6.4.3 Student Maintenance for FHCT1014

```
C:\Windows\py.exe
-----
Courses Student Maintenance
-----
FHCT1014 (INTRODUCTION TO DATA ANALYTICS)--->
2200012 John Anthony
2200011 Jamie Crossby
2210001 Tan Yin Jing
2200021 Wong Ai Fen
2200022 Mohd Hariz
2200023 Mark Adam
2200014 Daniel
2200015 Adrian Wong
2200016 Soo Ai Ling
2200024 Elizabeth Tan
-----
<A>dd <U>pdate <D>delete <E>xit : A
Student ID (YY####) <Q>uit >> 2200025
Student Name <Q>uit >> Chris Jordan
```

6.5 Sample of Text File for Marks Entered

FHCT1024 (PROGRAMMING CONCEPTS AND DESIGN)								

StudentID , T1(50/0.10) , T2(50/0.20) , A (100/0.20) , F (100/0.50)								

2200012	,	45	,	30	,	85	,	60
2200011	,	47	,	33	,	83	,	66
2210001	,	40	,	29	,	83	,	58
2113001	,	45	,	44	,	85	,	88
2217033	,	48	,	35	,	85	,	70
2200013	,	50	,	37	,	83	,	74
2200014	,	50	,	35	,	80	,	70
2200015	,	42	,	25	,	80	,	50
2200016	,	43	,	37	,	80	,	74
2200017	,	50	,	38	,	80	,	76

6.6 Student Results

6.6.1 Student Results for FHCT1024

C:\Windows\py.exe

```
-----
FHCT1024 (PROGRAMMING CONCEPTS AND DESIGN)
-----
2200012  John Anthony           68  B
2200011  Jamie Crosby             72  B+
2210001  Tan Yin Jing             65  B
2113001  Chia Xi Xin              88  A
2217033  Ahmand Ali               76  A-
2200013  Natasha                  78  A-
2200014  Daniel                   75  A-
2200015  Adrian Wong              59  C+
2200016  Soo Ai Ling              76  A-
2200017  Desmond Tan Sok Man      79  A-
-----
```

6.6.2 Student Results for FHCT1014

```
C:\Windows\py.exe

-----
FHCT1014 (INTRODUCTION TO DATA ANALYTICS)
-----
2200012   John Anthony           73   B+
2200011   Jamie Crossby              74   B+
2210001   Tan Yin Jing               65   B
2200021   Wong Ai Fen               79   A-
2200022   Mohd Hariz                56   C+
2200023   Mark Adam                 86   A
2200014   Daniel                   80   A
2200015   Adrian Wong             77   A-
2200016   Soo Ai Ling              80   A
2200024   Elizabeth Tan            84   A
2200025   Chris Jordan             86   A
-----
```

6.7 Data Validation

6.7.1 Data Validation 1

```
C:\Windows\py.exe

-----
Course Assessment Systemc -> Process steps
-----
<1>Config <2>Generate <3>Upload <4>Compute Final <E>xit : 6
Error in option entered
_
```

6.7.2 Data Validation 2

```
C:\Windows\py.exe

Report (listing of final marks) --->
[FHCT1024, FHCT1014, FHMM1034, FHSC1134]
Enter course code <Q>uit >> FHCT1012
Invalid course code entered
```

~ END OF GUIDELINES ~