

**1.0 Module Learning Outcomes:**

- Construct a programmable solution using appropriate problem-solving methods and programming concepts to given scenario. (C3, PLO2)

**2.0 Coursework Learning Outcomes:**

At the end of this coursework, you should be able to:

- Create coding-based solution for small-scale problems.
- Explain programming concepts and apply them using Python programming language.
- Implement basic data storage using text file.

**3.0 Type**

- Group Assignment

**4.0 Group Assignment Description****Tuition Centre Management System**

Excellent Tuition Centre (ETC) is one of leading tuition centre in city providing coaching classes for secondary school students (Form 1 – Form 5) for all subjects. This centre decided to enhance their business by having a management system to handle information of their students and subjects offered. ETC requires you to develop a Python program to simulate this case study containing following features:

**Overview of System:**

There are 4 types of registered users which are Admin, Receptionist, Tutor, and Student. The system should have a login feature to allow the user to key in username/email and password only. Maximum three attempts of login are allowed. The system should identify role of the user and redirect to menu option respectively.

Following are the functionalities of each role:

**1. Admin**

- Register and delete tutors. Assign tutors to respective levels and subjects.
- Register and delete the receptionist.
- View monthly income report based on level and subject.
- Update own profile.

## 2. Receptionist

- Register a student and enroll the student in up to 3 subjects. During enrolment, record student information for eg: name, IC/Passport, email, contact number, address, level, subjects, the month of enrolment, etc (you may add other relevant information)
- Update subject enrollment if students request to change subject.
- Accept payment from students and generate receipts.
- Delete students who have completed their studies.
- Update own profile.

## 3. Tutor

- Add class information (e.g. subject name, charges, class schedule, etc) .
- Update and delete class information.
- View the list of students enrolled in his/her subjects.
- Update own profile.

## 4. Student

- View the schedule of his/her classes.
- Send a request to the receptionist to change the enrolled subject.
- Delete the request (which is still pending) sent to the receptionist to change the subject.
- View payment status with the total balance that needs to be paid, if any.
- Update own profile.

### **IMPORTANT INSTRUCTIONS:**

*This is a group assignment. Upon submission of your assignment, you are required to present your assignment at a date and time specified by your module lecturer.*

*Each team member is required to contribute towards some features in the system & documentation, present and explain his or her work accordingly. Each team member should be able to answer questions during presentation with regards to the overall systems project and or specific question(s) related to the codes used in the development.*

## 5.0 Requirements

- i. You are required to carry out extra research for your system and document any logical assumptions you made after the research.
- ii. Your program should use symbolic constants where appropriate. Validations need to be included to ensure the accuracy of the system. State any assumptions that you make under each function.
- iii. You are required to store all data in text files. There is no limit on the number of text files that can be used but they should be kept minimum.
- iv. You are expected to use list and functions in your program. Your program must embrace modular programming technique and should be menu-driven.
- v. You may include any extra features which you may feel relevant and that add value to the system.
- vi. There should be no need for graphics in your program, as what is being assessed, is your programming skill not the interface design. The marking scheme for the assignment has been provided so that you clearly know how the assessment for this assignment would be done.
- vii. You should include good programming practice such as comments, variable naming conventions and indentation.
- viii. In a situation where a student:
  - *Failed to attempt the assignment demonstration, overall marks awarded for the assignment will be adjusted to 50% of the overall existing marks.*
  - *Found to be involved plagiarism, the offence and will be dealt in accordance to APU regulations on plagiarism.*
- ix. You are required to use Python programming language to implement the solution. Use of any other language like C/C++/Java is **not allowed**.

## 6.0 Deliverables

You are required to submit a softcopy of:

- i. Program coded in Python – submitted as .py file.
- ii. Text files created through test data – submitted as .txt files.
- iii. A documentation of the system – submitted as pdf file - that incorporates basic documentation standards such as header and footer, page numbering and includes:
  - Cover page – include all group member name & TP number
  - Table of contents
  - Introduction and assumptions
  - Design of the program – using pseudocode **or** flowcharts – which adheres to the requirements provided above
  - Explanation of programming concepts applied with sample segment of source code from the system created.
  - Additional features source code with explanation (if any)
  - Screenshots of sample input/output with explanation
  - Conclusion
  - References using APA referencing format

## 7.0 Performance Criteria

### **Distinction (80% and above)**

This grade will be assigned to work which meets all of the requirements stated in the question. The program runs smoothly when executed. There is clear evidence and application of Python concepts up to advanced level. The program solution is unique with excellent coding styles and validation. The program implemented maps completely against the design (pseudocode and flowchart) as seen in the documentation. The design of the solution varies in styles and has unique logic with hardly any errors / omissions. The documentation does not have any missing components. Sample inputs/outputs documented have clear explanation. Student must be able to provide excellent explanation of the codes and work done, show additional concepts / new ideas used in the solution, able to answer all questions posed with accurate / logical answers / explanation provided with sound arguments and clear discussion. Overall an excellent piece of work submitted.

### **Credit (65%-74%)**

This grade will be assigned to work which is considered to be of good standard and meets most of the requirements stated in the question. The program runs smoothly when executed. There is clear evidence and application of Python concepts up to at least intermediate level. The program solution is unique with good coding styles and validation. The program implemented maps well against the design (pseudocode and flowchart) as seen in the documentation. The design of the solution varies in styles and has unique logic with minor errors / omissions. The documentation does not have any missing components. Sample inputs/outputs documented with some explanation. Student must be able to provide good explanation of the codes and work done, answer most questions posed with mostly accurate / logical answers / explanation. Overall a good assignment submitted.

### **Pass (50%-64%)**

This grade will be assigned to work which meets at least half of the basic requirements (approximately 50%) stated in the questions. The program runs smoothly when executed. There is clear evidence and application of Python concepts at basic level. The program solution is common with basic coding styles and validation. The program implemented somewhat maps with the design (pseudocode and flowchart) as seen in the documentation. The design of the solution is average in terms of logic and style with some errors / omissions. The documentation

has some missing components. Sample inputs/outputs documented but without any explanation. Student must be able to explain some codes and work done and able to answer some questions posed with some accurate / logical answers / explanation. Overall an average piece of work submitted.

**Fail (Below 50%)**

This grade will be assigned to work which achieved less than half of the requirements stated in the question. The program is able to compile but not able to execute or with major errors. The program solution has only basic coding styles with no validation. The program solution has little or no mapping with the design. The design of the solution has major / obvious errors / omissions. The documentation has some missing essential components. Student is barely able to explain the codes / work done and answer given on the questions posed but with mostly inaccurate / illogical answers / explanation. Overall a poor piece of work submitted.