Course Learning Outcomes:

Upon completion of this assignment you should be able to:

CLO1	Translate simple problem statements into programmable solutions using flow chart/pseudo code (C3, PLO2).
CLO2	Comprehend knowledge of basic and advanced programming concepts (C2, PLO1).
CLO3	Show the ability to write computer programs for a given problem statement (P4, PLO3).

1.0 INDIVIDUAL ASSIGNMENT DESCRIPTION

REAL CHAMPIONS SPORT ACADEMY SYSTEM (RCSAS)

REAL CHAMPIONS SPORT ACADEMY is one of the fast-growing Sport Centre in Malaysia that employed tens of coach to conduct sport classes on Swimming, Badminton, Football, Archery, Gymnastics, Volleyball, Basketball, Cricket, Tennis and Table Tennis in their various Sport Centres. All the coaches are paid by hourly rate ranged from RM100.00 – RM500.00, depends on the sport they give training. Coaches can be rated by the students on their training performance with a scale rating from 1-5, where 1 is "very poor performance" and 5 is "excellent performance". The Sport Centre needs a system to store information about their coaches (identified by Coach ID, Name, Date Joined, Date Terminated, Hourly Rate, Phone, Address, Sport Centre Code, Sport Center Name, Sport Code, Sport Name, Rating [a single digit integer number ranging from 1 to 5]). To maintain quality and consistency, one coach is allowed to give training on the sport in their specialized sport only.

Sport Centers are located in various locations and each of the Sport Centre is managed by an admin staff. All the sports' record, coaches' record and students' record are managed and maintained by the Sport Centre Head Quarter Human Resources Department. The current filing system of records is by categorizing the coaches by their respective location. i.e. all coaches' records are filed together in the Sport centre location. In each location, the coach's records are filed alphabetically by the coach's name. The main problem with this filing system is when searching for a record, the record could not be easily found as it might have been misplaced in another Sport Centre or the record is not placed in the correct alphabet sequence. This problem proves challenging for the Sport Center admin staff when wanting to retrieve a coaches' record, sports' record and students' record quickly.

On the other hand, students need to register online by exploring system and get information about Sport Center, Sports, Coach, Coach performance with stars, Sport fees, Schedule of Sport, etc.

The admin manager of the Sport Center has decided that it is time to computerize the records of Coaches serving, Sports and students in the Sport Center due to the problems with the manual filling system. Since you have some knowledge in developing a computerized system, the Sport Center HR Manager has approaches you to assist them in developing the system.

This project requires you to develop Python program for the Sport Academy System which have 2 users and should contains features stated below:

Functionalities of Admin

i. Login to Access System.
ii. Add Records of
    a. Coach
    b. Sport
    c. Sport Schedule
iii. Display All Records of
    a. Coach
    b. Sport
    c. Registered Students
iv. Search Specific Records of
    a. Coach by Coach ID
    b. Coach by overall performance (Rating)
    c. Sport by Sport ID
    d. Student by Student ID
v. Sort and display Record of
    a. Coaches in ascending order by names.
    b. Coaches Hourly Pay Rate in ascending order
    c. Coaches Overall Performance in ascending order
vi. Modify Record of
    a. Coach
    b. Sport
    c. Sport Schedule
vii. Exit

Functionalities of All Student (Registered / Not-Registered)

i. View details of
    a. Sport
    b. Sport Schedule
ii. If new student Register to Access other Details
iii. Exit

Functionalities of Registered Student

i. Login to Access System
ii. View Detail of
    a. Coach
    b. Self-Record
    c. Registered Sport Schedule only
iii. Modify Self Record
iv. Provide feedback and Star to Coach.
v. Exit

2.0 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 the 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.

x. Global variables, build in functions like min, max, sort, etc… are not allowed.

3.0 DELIVERABLES

You are required to submit a softcopy of:

i. Program coded in Python – submitted as .py file.
   - Name the file under your name and TP number (e.g. KATHY_SIERRA_TP123456.py)
   - Start the first two lines in your program by typing your name and TP number (e.g. as follows):
     #KATHY SIERRA
     #TP123456

ii. Text files created through test data – submitted as .txt files.

iii. A documentation of the system – submitted as NAME_TPNUMBER.pdf file - that incorporates basic documentation standards such as header and footer, page numbering and includes:
   - Cover page
   - Table of contents
   - Introduction and assumptions
   - Design of the program – using pseudocode **and** flowcharts – which adheres to the requirements provided above
   - Program source code and explanation
   - Screenshots of sample input/output and explanation
   - Conclusion
   - References (if any) using Harvard Name Referencing

4.0 ASSESSMENT CRITERIA

i. Design (Pseudocode and Flowchart) 30%

Detailed, logical and accurate design of programmable solution.

ii. Coding / Implementation (Python code) 30%

Application of Python programming techniques (from basic to advance); good programming practices in implementing the solution as per design; and adequate validation meeting all system requirements with all possible additional features.

iii. Documentation 25%

Adherence to document standard format and structure; screen captures of input/output with explanation; and inclusion of generated text files.

iv. Demonstration 15%

Ability to run, trace code, explain work done and answer questions.

5.0 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.