Course Syllabus

COMP 8080SEF Python Programming
Effective Period: 2024 Autumn Term -

1. Course Data

Credit-units	: 3
Medium of Instruction	: English
Contact Hours	: 39
Pre-requisite	: None
Excluded combination	: COMP S258F, COMP S258, COMP 1080SEF

2. Introduction

2.1 COMP 8080SEF is a face-to-face, three-credit course offered by the School of Science and Technology. It is a core course in the full-time face-to-face Master of Computing programme.

3. Aims and Course Intended Learning Outcomes (CILOs)

3.1 Aims

The aim of this course is to develop in students the fundamental knowledge and skills in computer programming.

3.2 CILOs

Upon completion of this course, students should be able to:

1. *Write* computer programs in a structural programming language.
2. *Write* programs with structures, arrays/lists, and function calls.
3. *Design* and *develop* structured programs to solve problems.

4. Course Contents

4.1 The course consists of the following topics:

Topic 1: Introduction to the Python programming

- Essentials of computer programming
- Setting up the development environment
- Running Python programs
- Values, variables, data types, operators
- Basic input and output
- Basic file input and output

Topic 2: Control structures

- Conditional structures
- Repetition structures

Topic 3: Python functions

- Introduction to functions
- Function definition and parameters
- Parameter passing and return values

Topic 4: Data collections

- Python List and String
- Python Dictionary
- Concepts of mutable and immutable

5. Teaching and Learning Strategy

5.1 There will be 2 hours of lecture and 1 hour of tutorial each week. Lectures are used to explain concepts in programming and demonstrate the process of programming. Computer laboratory sessions allow students to practice what they have learned with relevant programming problems and challenges.

5.2 The teaching plan is shown below:

Content Topics	*Lecture hours*	*Tutorial hours*
1. Introduction to the Python programming	6	3
2. Structured programming	8	4
3. Problem solving with programming	4	2
4. Advanced Topics	8	4
Total	26	13

6. Assessment

6.1 Course assessment includes continuous assessment and a final examination.

- Continuous assessments will account for 50% of the Overall Course Score (OCS). Continuous assessment includes a test and laboratory exercises. The test involves multiple-choice questions and short questions. The laboratory exercises assess students' programming skills.
- The course project evaluates students' programming capacity and problem-solving ability by identifying a real-life problem and providing a solution with Python programming.

6.2 The weighting of the assessment components are shown below.

Assessment	*Weighting*
a. Laboratory exercise	20%
b. Test	30%
c. Individual Course Project	50%

7. Alignment of Course Contents and Assessments with CILOs

7.1 The learning outcomes, content topics and assessments are related as follows:

CILOs	*Content Topics*	*Assessments*
1. *Write* computer programs in a structural programming language.	1, 2, 3, 4	a, b, c
2. *Write* programs with structures, lists, and function calls.	1, 2, 3, 4	a, b, c
3. *Design* and *develop* structured programs to solve problems.	1, 2, 3, 4	a, b, c

8. Textbook and References

8.1 Textbook

Gaddis, T. (2019) Starting Out with Python, 4th edn, New York NY: Pearson