UNIVERSITY OF MAURITIUS
MODULE CATALOGUE

1. GENERAL INFORMATION

Academic Year: 2024-2025

Semesters: 1 and 2

Module Name	Codes	Duration (hours)		No. of credits
Computer Programming	ICDT1201Y	Direct Contact ● Lectures ● Practicals ● Tutorials	60	12 LCCS credits
		Self-Learning	120	
		Other Learning Activities*	180	

*working on assignments, sitting for Class Tests and preparation time for same, sitting for Examinations and preparation time for same, group work, attending Workshops/Conference recommended by the Department/Faculty, fieldwork, site visits/trips, additional practicals, presentations among peers, experiential learning, placements/internships and guest lectures.

2. PRE-REQUISITE(S)/PRE-REQUIREMENT(S)

None.

3. AIMS

The core of any computing program is problem solving. Programmers must first understand how a human solves a problem, then translate this understanding into something a computer can do, and finally implement the solution using a specific programming language to complete the task. Therefore, an introductory module (in problem solving techniques, computer programming, data structures and object-oriented concepts) is a core component of any first-year computing course and is an essential prerequisite to many other modules that follow in the second and third year. In this module we aim at enabling the student to develop programming skills. After completing this module, students should have acquired the fundamental programming skills for problem solving and software development.

4. OUTLINE SYLLABUS

Through Contact Hours	Introduction to Computers & Programming, Problem Solving, Algorithm and Design, Data Types, Syntax and Semantics, Input and Output Operations, String Manipulation, Control structures, Arrays/Lists, Modular Programming, File input and output, Data Structures, Object Oriented Concepts, Using in-built functions.
Through Self-Learning	Solving computational problems requiring complex decision Structures, Program Documentation, Program Testing, Application of Software Programming Ethics.

5. LEARNING OUTCOMES AND ASSESSMENT CRITERIA

Having studied this module, the students should be able to achieve the following learning outcomes. The assessment criteria used to reflect the expected learning outcomes are also given hereunder:

	Learning Outcomes	Assessment Criteria
1	Demonstrate a sound understanding of problem-solving techniques and algorithm design.	● Recognize the importance of problem solving ● Explain common terminology used in problem solving ● Identify the different tools used in problem solving ● Recognize the difference between an algorithm and a program ● Identify the various symbols used in flowcharts
2	Demonstrate an understanding of data types, variables, syntax and semantics	● Understand the concepts literals, identifiers and mathematical operations ● Understand the difference between a valid and an invalid identifier ● Distinguish between different types of data ● Use variables and write simple expressions
3	Define the key components of a statement in a program.	● Understand the construct of simple statements such as assignments and function calls ● Practice the use of mathematical operators in program statements

		● Differentiate between input and output statements
4	Explain the concept of working with text	● Understand how to create strings of characters ● Understand the concept of string when printing information ● Use the String data type and associated functions
5	Introduce the concepts of Arrays/Lists	● Understand the importance of Arrays/Lists as an important data structure to store multiple data elements
6	Understand the different control structures	● Understand what is a Boolean Type and how it can used to choose which statements to execute ● Understand the if ,if else, if else if else and nested if statements ● Comprehend the while loop and the for loop ● Write simple programs using different conditional structures and loops.
7	Define the principles of functions	● Explain the importance of organizing a program using functions ● Write simple functions ● Recognize the importance of parameters in functions and return statement ● Understand that a function can call other functions and can have branching statements ● Write recursive functions
8	Demonstrate the use of text Files	● Write programs that can read from and write to text files ● Differentiate between the read, write and append modes for opening files ● Write complex programs that can manipulate data
9	Introduction to Object-Oriented Programming	● Understand the concepts of Object-oriented programming using the concepts by creating a Class, its attributes and methods ● Understand the reasoning behind object creation and operator overloading ● Write simple object-oriented programs

10	Introduce the concepts of advanced data structures	● Understand the difference between commonly-used data structures ● Write medium-sized programs that use different data structures to manipulate data
11	Demonstrate the principles of Sorting and Searching algorithms	● Introduce the concepts of searching and sorting ● Use in-built searching algorithms ● Use in-built sorting algorithms ● Write medium-sized programs that use basic searching and sorting algorithms

6. COORDINATORS

	Programme Coordinator	Module Coordinator
Name		Mr Selvanaden SATHAN
Department		I.C.T
Building		Phase 2
Room Number		2.17.5
Phone No.		4037748
E-mail address		s.sathan@uom.ac.mu
Consultation Time		Email to arrange meeting

7. LECTURER(S)

ICT Department

Name	Mrs. Vidasha Ramnarain-Seetohul	Mr Selvanaden SATHAN	Associate Professor (Dr) Sameerchand Pudaruth
Department	ICT	I.C.T	ICT
Building	Phase 2	Phase 2	Phase 2/CITS/FoICDT
Room Number	2.5	2.17.5	2.24
Phone No.	4037892	4037748	403 7754
E-mail address	v.seetohul@uom.ac.mu	s.sathan@uom.ac.mu	s.pudaruth@uom.ac.mu
Contact Hours	As per Timetable	As per Timetable	As per Timetable

Consultation Time	email for appointment		Email for appointment
Contact Address (for P/T)			

SIS Department

Name	Zarine Cadersaib	Leckraj Nagowah
Department	SIS	SIS
Building	Engineering Tower	Phase II Building
Room Number	3.6	2.17
Phone No.	403 7829	4037742
E-mail address	z.cadersaib@uom.ac.mu	l.nagowah@uom.ac.mu
Contact Hours	As per time table	As per time table
Consultation Time	On request	Email for appointment
Contact Address (for P/T)		

DT Department

Name	Roushdat Elaheebocus	
Department	Digital Technologies	

Building	Phase II	
Room Number	2.18D	
Phone No.	4037877	
E-mail address	r.elaheebocus@uom.ac.mu	
Contact Hours	As per time table	
Consultation Time	Monday 7:30am-9am	
Contact Address (for P/T)		

8. VENUE AND HOURS/WEEK

Lectures may be held in a blended mode and practicals will be done face-to-face

Hours/week: 3 hours direct contact through lectures, practicals and tutorials for ten weeks per semester.

More details in Section 9 Module Map

9. MODULE MAP

Week	Topics	Units	DC	SS	OA
1	**Introduction to Computers & Programming**	● Computer Model ● Computer Programs ● Translation	**3**	**3**	**3**
2	**Algorithm and Design - Problem Solving Concepts**	● Problem Solving ● Abstraction ● Concept of Algorithms ● Program Design using flowchart	**3**	**3**	**3**
3	**Data Types, Syntax and Semantics**	● Syntax ● Semantics ● Variables ● Arithmetic Operations ● Expressions and Assignments	**3**	**6**	**6**
4	**Input and Output Operations**	● Functions to carry out IO	**3**	**6**	**10**
5	**Conditional Structures 1**	● Selection (in flowcharts) ● Relational Operators ● Simplex Boolean Expression ● if, if-else	**3**	**6**	**11**
6	**Conditional Structures 2**	● Boolean Operators ● Boolean Logic ● Building Complex Boolean Expressions ● multi-way if-else	**3**	**6**	**11**
7	**Iterative Structures**	● Loops (in flowcharts) ● Type of Loops	**3**	**6**	**11**
8	**Arrays/Lists**	● Defining Arrays ● Indexing ● Array Manipulation (I/O)	**3**	**6**	**10**
9	**Functions**	● Importance of Functions ● Writing Functions ○ Passing Parameters to Functions ○ Returning Values	**3**	**9**	**12**
10	**Test1**				

11	**Recursion**	● Definition ● Control Flow in Recursive functions ● Writing Simple Recursive Functions	**3**	**6**	**10**
12	**Return of Test1**		**0**	**3**	**3**
			30	**60**	**90**
END OF SEMESTER ONE					
13	**Functions and Lists**	● Functions which manipulate lists	**3**	**6**	**10**
14	**String Manipulation**	● Strings ● Using Strings and in-built functions	**3**	**6**	**8**
15	**File I/O**	● Importance of Files ● Writing to Files ● Reading from Files	**3**	**6**	**8**
16	**Data Structures 1**	● Data Structures ● Lists ('Advanced' Operations of List (e.g Searching, Sorting, Insertions and Removal)	**3**	**6**	**9**
17	**Data Structures 2**	● Stacks and Queues	**3**	**6**	**10**
18	**Data Structures 3**	● Dictionaries ● Problem Solving using ADT	**3**	**6**	**12**
19	**Test 2**				
20	**Object Oriented Concepts 1**	● The Object Oriented Paradigm ● Terminologies ○ Classes ○ Objects Attributes	**3**	**7**	**9**
21	**Object Oriented Concepts 2**	Classes ○ Methods ○ Constructors ● Access Modifiers ● Inheritance	**3**	**5**	**6**
22	**Defining Classes**	● Writing Classes ● Defining Objects	**3**	**6**	**9**

23	**Return of Test2 and Assignment Presentation**		**1**	**2**	**3**
24	**Revision**		**2**	**4**	**6**
			60	**120**	**180**
END OF SEMESTER TWO					

Abbreviations: ***DC****: Direct Contact;* ***SS****: Self Study;* ***OA****: Other Learning Activities*

DC *includes* ***L****: Lectures,* ***P****: Practicals,* ***T****: Tutorials;*

10. RECOMMENDED BOOKS/JOURNALS/WEBSITES

Reading Material:

- Python TutorialsPoint, Available at: https://www.tutorialspoint.com/python/index.htm

Reference Books:

- Practical Programming: An Introduction to Computer Science Using Python 3, by Paul Gries, by Jennifer Campbell and Jason Montogo.
- Introduction to Computation and Programming Using Python: With Application to Understanding Data (The MIT Press), by John V. Guttag

11. TEST(S)/ASSIGNMENT(S)/PRACTICAL(S)

Semester	Title	Max Marks
1	Class Test (WK10) Topics to be assessed: Week 1 - Week 8	15
2	Class Test (WK19) Topics to be assessed: Week 1 - Week 15	15
3	Practicals	5
4	Assignment Demonstration (WK23) Topics assessed: All Topics	15

12. ASSESSMENT

(i) Written Examination

Paper Structure	
Sections (if any): None	No. of questions to be answered: 4
Paper Duration:	3 hours
Weighting (%): 50	
Total Marks: 100	Pass Mark: 40

(ii) Continuous Assessment

	Weighting (%)
Assignment(s):	15.00%
Practical(s) (Lab based tutorials):	5.00%
Test(s):	30.00%
Total Marks:	50.00%

13. OFFICE HOURS

Email to arrange a meeting.

14. PORTFOLIO REQUIREMENT

All students should keep a portfolio of all coursework for their respective Programme of studies and should be made available upon request, to the Faculty/Centre Examination Office.

15. OTHER INFORMATION

- **Independent Learning.** At the end of each topic, you will be provided with additional exercises that you will need to do in your own time after consulting your notes and other sources such as:
    - ❖ FreeCodeCamp (https://www.freecodecamp.org),
    - ❖ TutorialsPoint (https://www.tutorialspoint.com/python/index.htm).
- **Labs**. You will have one hour of formal laboratory time per week. During this time you will be expected to come with your solutions for the respective lab questions and demonstrate the same. The tutor(s) in attendance will be in the lab to help you out on problems which you would have encountered while working on your weekly lab sheets (which are expected to have been worked on prior to coming to the lab) . Lab-based tutorials will be assessed and contribute marks towards your continuous assessment and ultimately to your final grade for the module.
- **Group work**. Teamwork is encouraged to have a better understanding of the topics.
- **Plagiarism**. Plagiarism is a serious academic offense and appropriate disciplinary actions will be taken as per UoM rules. For more information check the following link .

16. APPROVAL BY HEAD OF DEPARTMENT

Module Catalogue approved at Departmental Meeting on	
Head of Department **Name and Signature**	

A copy of the approved Module Catalogue has to be submitted to the relevant Dean of Faculty for records purposes.