City University of Hong Kong
Course Syllabus

offered by Department of Computer Science
with effect from Semester A 2020/21

Part I Course Overview

Course Title:	Introduction to Computer Programming
Course Code:	CS1302
Course Duration:	One semester
Credit Units:	3 credits
Level:	B1
Proposed Area: *(for GE courses only)*	☐ Arts and Humanities ☐ Study of Societies, Social and Business Organisations ☐ Science and Technology
Medium of Instruction:	English
Medium of Assessment:	English
Prerequisites: *(Course Code and Title)*	Nil
Precursors: *(Course Code and Title)*	Nil
Equivalent Courses: *(Course Code and Title)*	Nil
Exclusive Courses: *(Course Code and Title)*	CS1102 Introduction to Computer Studies

Part II Course Details

1. Abstract

(A 150-word description about the course)

This course aims to introduce to students with key concepts, techniques, and good practices of programming using a high-level programming language such as Python.

2. Course Intended Learning Outcomes (CILOs)

(CILOs state what the student is expected to be able to do at the end of the course according to a given standard of performance.)

No.	CILOs[#]	Weighting* (if applicable)	Discovery-enriched curriculum related learning outcomes (please tick where appropriate)		
			A1	A2	A3
1.	Explain the structure of a computer program.	10%	✓	✓	
2.	Analyze, test and debug computer programs.	20%	✓	✓	
3.	Apply proper programming techniques to solve a task.	50%		✓	
4.	Construct well-structured programs.	20%		✓	✓
** If weighting is assigned to CILOs, they should add up to 100%.*		100%			

[#] Please specify the alignment of CILOs to the Gateway Education Programme Intended Learning outcomes (PILOs) in Section A of Annex.

A1: Attitude
Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.

A2: Ability
Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to self-life problems.

A3: *Accomplishments*
Demonstrate accomplishment of discovery/innovation/creativity through producing /constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

3. Teaching and Learning Activities (TLAs)

(TLAs designed to facilitate students' achievement of the CILOs.)

Teaching pattern:
Suggested lecture/laboratory mix: 2 hrs. lecture; 2 hrs. laboratory.

TLA	Brief Description	CILO No.				Hours/week (if applicable)
		1	2	3	4	
Lecture	Various programming concepts and techniques will be introduced, explained and demonstrated with examples.	✓	✓	✓	✓	
Lab	The laboratory sessions are designed to enable the students to put theory into practice and be proficient in a programming language. Besides short tutorial exercises, students can also try out their programs during the labs using a common integrated development environment. Feedback will be given to students on their work.	✓	✓	✓	✓	
Assignment	The assignments are more challenging tasks compared with laboratory exercises. The students need to analyze the requirements and design programming solutions by applying and combining various techniques learnt from classes. They are also required to implement their solutions as practical computer programs and to explain their ideas/algorithms using suitable presentation methods (e.g., a report, flowchart, etc.).		✓	✓	✓	

4. Assessment Tasks/Activities (ATs)

(ATs are designed to assess how well the students achieve the CILOs.)

Assessment Tasks/Activities	CILO No.				Weighting*	Remarks
	1	2	3	4		
Continuous Assessment: 50%						
Quiz	✓		✓	✓	20%	Correctly explain different parts of a computer program and the behaviour of its execution. Find out program errors and make corrections. Apply proper programming techniques to solve a task. Construct well-structured programs.
Assignment		✓	✓	✓	30%	Individual or group assignments on program development and testing.
Examination^: 50% (duration: 2 hours)						
** The weightings should add up to 100%.*					100%	

^ For a student to pass the course, at least 30% of the maximum mark for the examination must be obtained.

5. Assessment Rubrics

(Grading of student achievements is based on student performance in assessment tasks/activities with the following rubrics.)

Assessment Task	Criterion	Excellent (A+, A, A-)	Good (B+, B, B-)	Fair (C+, C, C-)	Marginal (D)	Failure (F)
1. Quiz	1.1 ABILITY to explain, analyse and debug the structure of a computer program	High	Significant	Moderate	Basic	Not even reaching marginal levels
2. Assignment	2.1 CAPACITY for applying programming techniques	High	Significant	Moderate	Basic	Not even reaching marginal levels
3. Examination	3.1 CAPACITY for analyzing and writing effective computer programs	High	Significant	Moderate	Basic	Not even reaching marginal levels

Part III Other Information (more details can be provided separately in the teaching plan)

1. Keyword Syllabus

(An indication of the key topics of the course.)

The development of algorithms. Program design. Programming language. Control structures. Data types. Arrays/matrices, functions and parameters. Composite data types. List comprehension. Mathematics libraries. Structured decomposition. Programming style. Program testing. Introduction to recursion. Fundamentals on computer hardware architecture. Applications of programming to mathematical problem-solving methods.

Syllabus

1. Program development environment
   Software development process. Development tools. Program development environments.

2. Programming techniques and the development of algorithms
   Modular decomposition and stepwise refinement, principles of abstraction. Algorithms, the realisation of algorithms as programs. Program design: programming language, procedural abstraction, parameter-passing, control structures, iteration, recursion.

3. Data structures
   The concept of data types. Simple data types. Arrays/matrices. List comprehension. Strings. Composite data types.

4. Program development practice
   Elements of programming style. Program testing. Program documentation.

2. Reading List

2.1 Compulsory Readings

(Compulsory readings can include books, book chapters, or journal/magazine articles. There are also collections of e-books, e-journals available from the CityU Library.)

1.	Richard L. Halterman (2019). Fundamentals of Python Programming
2.	Allen Downey (2015). Think Python – How to Think Like a Computer Scientist. 2nd ed
3.	Eric Matthes (2015). Python Crash Course: A Hands-On, Project-Based Introduction to Programming.
4.	Paul Barry (2016). Head First Python: A Brain-Friendly Guide 2nd ed

2.2 Additional Readings

(Additional references for students to learn to expand their knowledge about the subject.)

1.	A Byte Of Python, by C.H. Swaroop Available online at: https://python.swaroopch.com/