City University of Hong Kong
Course Syllabus

offered by Department of Computer Science
with effect from Semester A 2021/22

Part I Course Overview

Course Title:	Computer Programming
Course Code:	CS2315
Course Duration:	One semester
Credit Units:	3 credits
Level:	B2
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)*	CS2310 Computer Programming CS2311 Computer Programming
Exclusive Courses: *(Course Code and Title)*	CS2313 Computer Programming CS2360 Java Programming

Part II Course Details

1. Abstract

(A 150-word description about the course)

This course aims to equip the students with key concepts and techniques of programming using a high-level object-oriented programming language and to develop practical skills in producing quality programs. No prior programming or computer science experience is required.

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 an object-oriented computer program.	10%	✓	✓	
2.	Analyze, test and debug computer programs.	15%	✓	✓	
3.	Solve a task by applying effective programming techniques, which involve algorithm and data structures.	60%		✓	
4.	Design and construct well-structured programs with good programming practices.	15%		✓	✓
	** 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/tutorial/laboratory mix: 2 hours lecture; 2 hours 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. The laboratory exercises consist of programming tasks and students can try out their programs using a common integrated development environment. Feedback will be given to students on their work.	✓	✓	✓	✓	
Assignment	The assignments are more comprehensive tasks compared with laboratory exercises. The students need to consider the given requirements and design programming solutions by applying and combining various techniques learnt from lectures and laboratory exercises. Students are 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: 40%						
Quiz	✓		✓	✓	20%	Correctly explain the structure of an object-oriented computer program
Assignment		✓	✓	✓	20%	Select proper test cases to assess the correctness of a program Students are required to work on assignments at least once every four weeks
Examination^: 60% (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	ABILITY to explain, analyse and debug the structure of a computer program	High	Significant	Moderate	Basic	Not even reaching marginal levels
2. Assignment	CAPACITY for applying programming techniques	High	Significant	Moderate	Basic	Not even reaching marginal levels
3. Examination	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.)

Program design, development of algorithms, programming language, control structures, data types, arrays, file I-O, recursion, and data structures, object-based programming: data abstraction, classes, and the class library; programming style, program testing.

Syllabus:

1. Computers and programming
   Hardware/software hierarchy, the computer as a multi-level language machine. The software development process. Program development environments.

2. Programming techniques and the development of algorithms
   Algorithms, programming language, modular decomposition and procedural abstraction, automatic and dynamic variables, parameter-passing by reference and by value for atomic data, objects, and arrays, control structures, iteration.

3. Data structures
   The concept of data types. Simple data types. Arrays. Strings. Files. Data abstraction: encapsulation, information hiding. Defining and using classes. The class library. Data structures.

4. Program development practice
   Professional programming styles. 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 (2015). Fundamentals of C++ Programming. Southern Adventist University.

2.2 Additional Readings

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

1.	S.B. Lippman, J. Lajoie and B. Moo (2012). *C++ Primer.* Addison Wesley, 5th edition.
2.	H.M. Deitel & P.J. Deitel (2011). *C++ How to Program.* Pearson Int. Edition, 8th edition.
3	Walter Savitc (2010). *Absolute C++*. Addison-Wesley, 4th edition.