City University of Hong Kong
Course Syllabus

offered by Department of Computer Science
with effect from Semester B 2018/19

Part I Course Overview

Course Title:	Computer Programming
Course Code:	CS2310
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)*	CS1102 Introduction to Computer Studies or CS1302 Introduction to Computer Programming or equivalent
Equivalent Courses: *(Course Code and Title)*	CS2311 Computer Programming
Exclusive Courses: *(Course Code and Title)*	CS2313 Computer Programming

Part II Course Details

1. Abstract

This course aims to equip students with essential programming skills for object-oriented (OO) programming, so as to lay a solid foundation for other computer science related courses. After finishing this course, students should be able to articulate and write an object-oriented program, and have adequately grasped the necessary fundamental computer programming knowledge on which other computer science courses are built.

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.	Describe and explain the fundamental concepts, syntax, style and structure of object-oriented programs.	10%		✓	
2.	Create and develop solutions to simple problems / tasks and implement solutions with good programming practice in an object-oriented programming language.	60%			
3.	Trace, analyse, validate and deduce the output of given program code of different complexities.	15%	✓	✓	
4.	Demonstrate working knowledge on advanced programming features such as object initialization, inheritance, overloading and polymorphism.	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	
Laboratory	Laboratory exercises, consisting of programming problems, are designed for students to put theory into practice and be proficient in an OO programming language. Students are required to create computer programs using a program development tool that supports an OO programming language.	✓	✓	✓	✓	
Assignment	Assignments are intended to require students to solve more challenging problems compared with laboratory exercises. Students have to analyse the problems, break them down into manageable sub-problems, and apply (and possibly combine) various techniques learnt from lectures and laboratory exercises in order to create and develop solutions. Then they are required to implement the solutions as computer programs which conform to good programming practice, and to explain their solutions using suitable presentation methods (e.g., using a report, flowchart, etc.).	✓	✓	✓	✓	
Quiz	The quizzes provide opportunities for students to reflect on their understanding of various programming concepts and problem solving techniques. They are required to develop and implement programs or program segments to solve problems; and to predict and explain the behaviour of programs involving various programming concepts and techniques.	✓	✓	✓		

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%						
Assignments	✓	✓	✓	✓	25%	Students are required to work on assignments at least once every four weeks
Quiz	✓	✓	✓		25%	
Examination^: 50% (duration: 2 hours)						
** The weightings should add up to 100%.*					100%	

^ For a student to pass the course, at least 40% of the maximum mark for the continuous assessment and 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. Assignment and Quiz	ABILITY to WRITE PROGRAM to solve simple problems with good programming practices	High	Significant	Moderate	Basic	Not even reaching marginal levels
	DEMONSTRATE working knowledge on object oriented programming	High	Significant	Moderate	Basic	Not even reaching marginal levels
2. Examination	ABILITY to DESCRIBE and EXPLAIN the fundamental concepts, syntax, style and structure of object-oriented programs	High	Significant	Moderate	Basic	Not even reaching marginal levels
	ABILITY to WRITE PROGRAM to solve simple problems with good programming practices	High	Significant	Moderate	Basic	Not even reaching marginal levels
	ABILITY to TRACE and VALIDATE computer program	High	Significant	Moderate	Basic	Not even reaching marginal levels
	DEMONSTRATE working knowledge on object oriented programming	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.)

Programming language, program development, control structures, data types, arrays, files, recursion, pointers and dynamic variables. Object-oriented programming: class and object, inheritance, overloading, polymorphism. Scoping, programming style and tools.

Syllabus:

1. Computers and programming
   Hardware/software hierarchy, the computer as a multi-level language machine. Program development process and 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, arrays, control structures, iteration, recursion.

3. Data types and manipulation
   Concept of data types. Simple data types. Arrays. Strings. Files. Encapsulation and information hiding. Defining and using classes. Class library. Pointers. Input/output, arithmetic, logical and string operations.

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.)

	NA

2.2 Additional Readings

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

1.	H.M. Deitel & P.J. Deitel (2013). *C++ How to Program.* Pearson Int. Edition, 7th edition.
2.	Stanley B. Lippman, Josee Lajoie, Barbara E. Moo (2013). *C++ Primer.* Addison-Wesley, 15th edition.
3.	Chip Weems, Mark R. Headington, and Nell B. Dale (2013*). Programming And Problem Solving With C++:* Comprehensive, 6th Edition
4.	Walter Savitch (2014). *Problem Solving with C++,* 9th Edition