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:	Problem Solving and Programming
Course Code:	CS2312
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)*	CS2310 Computer Programming or CS2311 Computer Programming or equivalent
Equivalent Courses: *(Course Code and Title)*	Nil
Exclusive Courses: *(Course Code and Title)*	Nil

Part II Course Details

1. Abstract

(A 150-word description about the course)

This course aims to provide extensive practice in problem-solving using different programming paradigm, which includes the object-oriented programming, functional programming, and logic programming. Students will learn the fundamental concepts and distinctive features in these programming paradigms. They will develop skills to abstract data and entities from the problem domain, build models, design solutions using different paradigm principles and strategies, and implement solutions in these programs. Students will also explore tools and best practices in programming.

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.	Identify and describe fundamental programming paradigm concepts.	10%	✓	✓	
2.	Abstract data and entities from the problem domain, build models and design software solutions using different programming paradigm principles and strategies.	20%		✓	
3.	Implement the respective design of different programming paradigms in programs using a modern programming language to solve problems.	50%		✓	
4.	Apply tools and best practices in different programming paradigms.	10%	✓	✓	
5.	Evaluate and critique program coding and design based on different programming principles.	10%	✓		
	** 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 hrs. lecture; 1 hr. tutorial

TLA	Brief Description	CILO No.					Hours/week (if applicable)
		1	2	3	4	5	
Lecture	All CILOs will be introduced, explained, and demonstrated through lectures.	✓	✓	✓	✓	✓	
Tutorial	Students will practice with solving problems using pre-designed programs, helping them to gear up their ability and skills in all CILOs.	✓	✓	✓	✓	✓	
Quiz	The quiz will check students' achievement of the learning outcomes. This will provide timely feedback on their learning progress.	✓	✓	✓	✓	✓	
Assignments	The assignments will require students to solve challenging problems by designing and writing object-oriented programs. Assignments will serve as a learning and assessment tool.	✓	✓	✓	✓	✓	

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	5		
Continuous Assessment: 50%							
Quiz	✓	✓	✓	✓	✓	20%	
Assignments	✓	✓	✓	✓	✓	30%	Some portion may be allocated to weekly exercises
Examination^: 50% (duration: 2 hours)	✓	✓	✓	✓	✓	50%	
** 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. Assignments	1.1 ABILITY to articulate a convincing rationale for strategies used to design a solution for problem solving for different programming paradigms.	High	Significant	Moderate	Basic	Not even reaching marginal levels
	1.2 ABILITY to construct a program which conform to the program design and specification.	High	Significant	Moderate	Basic	Not even reaching marginal levels
	1.3 ABILITY to discover, explore and apply tools and best practices in different programming paradigms.	High	Significant	Moderate	Basic	Not even reaching marginal levels
	1.4 ABILITY to evaluate programs with a critical mind based on different programming paradigm principles.	High	Significant	Moderate	Basic	Not even reaching marginal levels
2. Quiz	2.1 ABILITY to identify and explain the concepts of different programming paradigms.	High	Significant	Moderate	Basic	Not even reaching marginal levels
	2.2 ABILITY to articulate a convincing rationale for strategies used to design a solution for problem solving with different programming paradigms.	High	Significant	Moderate	Basic	Not even reaching marginal levels
	2.3 ABILITY to construct a program which conform to the program design and specification.	High	Significant	Moderate	Basic	Not even reaching marginal levels
	2.4 ABILITY to discover, explore and apply tools and best practices in different programming paradigms.	High	Significant	Moderate	Basic	Not even reaching marginal levels
	2.5 ABILITY to evaluate programs with a critical mind based on different programming paradigm principles.	High	Significant	Moderate	Basic	Not even reaching marginal levels

Assessment Task	Criterion	Excellent (A+, A, A-)	Good (B+, B, B-)	Fair (C+, C, C-)	Marginal (D)	Failure (F)
3. Examination	3.1 ABILITY to identify and explain the concepts of different programming paradigms.	High	Significant	Moderate	Basic	Not even reaching marginal levels
	3.2 ABILITY to articulate a convincing rationale for strategies used to design a solution for problem solving with different programming paradigms.	High	Significant	Moderate	Basic	Not even reaching marginal levels
	3.3 ABILITY to construct a program which conform to the program design and specification.	High	Significant	Moderate	Basic	Not even reaching marginal levels
	3.4 ABILITY to discover, explore and apply tools and best practices in different programming paradigms.	High	Significant	Moderate	Basic	Not even reaching marginal levels
	3.5 ABILITY to evaluate programs with a critical mind based on different programming paradigm principles.	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.)

Problem solving in the object-oriented, functional programming, and logic programming paradigms.

Syllabus

1. Problem solving and programming paradigms
   Nature of problem solving. Structured programming. Object-based programming. Object-oriented programming, functional programming, logic programming.

2. Features of different programming paradigms
   Abstraction. Class. Encapsulation. Inheritance. Polymorphism. Functions. Relations. Datatypes. Recursive datatypes. Lambda calculus. Rules. Unification.

3. Constructing programs
   Association. Generalization. Specialization. Delegation. Realization. Aggregation. Dynamic Binding and Static Binding. Lazy evaluation. Recursion. Tail-recursion. Data abstraction.

4. Overview of programming languages
   Declarative, imperative and hybrid programming. General versus domain-specific languages. Translation from source to executable code: compilation, interpretation, intermediate code generation. Design and choice of programming paradigms and languages for problem solving.

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.	NONE

2.2 Additional Readings

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

1.	Object-Oriented Programming (OOP) in Python 3 https://realpython.com/python3-object-oriented-programming/
2.	David Mertz "Functional Programming in Python." O' 'Relly 2016
3.	David Barnes Object-Oriented Programming with Java: An Introduction Prentice Hall, 2000
4.	Richard L. Halterman Learning to Program with Python 2011
5.	Bruce Frederiksen Applying Expert System Technology to Code Reuse with Pyke 2008