TQF 3 Course Specifications

Section 1 General Information

1. Course code and course title

Thai	ICCS ๑๐๑ การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น
English	ICCS 101 Introduction to Computer Programming

2. Number of credits 4 (3-2-7) (Lecture/Lab/Self-study)
3. Program and type of subject
   - 3.1 Program: Bachelor of Science (Computer Science)
   - 3.2 Type of Subject: Required course
4. Course Coordinator and Course Lecturer
   - 4.1 Course Coordinator: Sunsern CHEAMANUNKUL, PhD
   - 4.2 Course Lecturers: Kritya BUNCHONGCHIT, PhD; Sunsern CHEAMANUNKUL, PhD
5. Trimester/ Year of Study
   - 5.1 Trimester: All trimesters (excluding summer session) / for all students in all International College Undergraduate Programs
   - 5.2 Course Capacity: Approximately 30 students per section
6. Pre-requisite: -
7. Co-requisites: -
8. Venue of Study: Mahidol University, Salaya Campus

Section 2 Goals and Objectives

1. Course Goals

To train students to write proper programs, preparing them to work with complex/larger codebase in the future.

2. Objectives of Course Development/Revision

2.1 Course Objectives

This course provides a hands-on introduction to problem-solving and computer programming, created using both TQF1 and ACM frameworks as a guideline.

2.2 Course-level Learning Outcomes: CLOs

By the end of the course, students will be able to (CLOs)

1. CLO 1 Recognize the role of algorithms in problem solving.
2. CLO 2 Describe the outcome and behaviors of control-flow constructs in imperative programs written in Python or similar languages.
3. CLO 3 Use and reason about data types, conditionals, iterations, functions, and recursion in the student's own code.
4. CLO 4 Select and suitably apply standard collections such as list, dictionary, etc.

5. CLO 5 Explain the mechanics of classes and object-oriented programming and implement simple logic in object-oriented style.
6. CLO 6 Write, test, and reason about small to medium-sized programs in Python.

Section 3 Course Management

1. Course Description

Role of algorithms in problem solving; Concepts of data types, including integers, floating-point numbers, and strings; Statements and expressions; Simple input/output; Conditionals and control-flow; Iteration, including loops and recursion; Functions; Basic collections, including resizable arrays and dictionaries; Classes and mechanics of object-oriented programming

บทบาทของขั้นตอนวิธีในการแก้ปัญหา; แนวคิดเกี่ยวกับชนิดของข้อมูลรวมถึงข้อมูลชนิดจำนวนเต็ม ตัวเลขชนิดจุดลอยตัว และสายอักขระ; ข้อความสั่งและนิพจน์; การนำเข้าและนำออกข้อมูลเบื้องต้น; เงื่อนไขและการควบคุมการทำงาน; การทำซ้ำ รวมถึงการวนซ้ำและรีเคอร์ชัน; ฟังก์ชัน; โครงสร้างเก็บข้อมูลเบื้องต้น รวมไปถึง แถวลำดับที่เปลี่ยนขนาดได้ ดิกชันนารี; คลาสและกลไกของการเขียนโปรแกรมเชิงวัตถุ

2. Credit hours per trimester

Lecture (Hour(s))	Laboratory/field trip/ internship (Hour(s))	Self-study (Hour(s))
36	24	84

3. Number of hours that the lecturer provides individual counseling and guidance.
1 hour/week

Section 4 Development of Students' Learning Outcome

1. Short summary on the knowledge or skills that the course intends to develop in students (CLOs)

By the end of the course, students will be able to:

1. CLO 1 Recognize the role of algorithms in problem solving.
2. CLO 2 Describe the outcome and behaviors of control-flow constructs in imperative programs written in Python or similar languages.
3. CLO 3 Use and reason about data types, conditionals, iterations, functions, and recursion in the student's own code.
4. CLO 4 Select and suitably apply standard collections such as list, dictionary, etc.
5. CLO 5 Explain the mechanics of classes and object-oriented programming and implement simple logic in object-oriented style.
6. CLO 6 Write, test, and reason about small to medium-sized programs in Python.

2. Teaching methods for developing the knowledge or skills specified in item 1 and evaluation methods of the course learning outcomes

ICCS101	Teaching methods	Evaluation Methods
CLO1	Reading assignment, interactive lecture, case studies, quiz, group activities, group discussion	Quiz, Homework, Examination
CLO2	Reading assignment, interactive lecture, case studies, quiz, group activities, group discussion	Quiz, Homework, Examination
CLO3	Reading assignment, interactive lecture, case studies, quiz, group activities, group discussion	Quiz, Homework, Examination
CLO4	Reading assignment, interactive lecture, case studies, quiz, group activities, group discussion	Quiz, Homework, Examination
CLO5	Reading assignment, interactive lecture, case studies, quiz, group activities, group discussion	Quiz, Homework, Examination
CLO6	Reading assignment, interactive lecture, case studies, quiz, group activities, group discussion	Quiz, Homework, Examination

Section 5 Teaching and Evaluation Plans

1. Teaching plan

Week	Topic	Number of Hours		Teaching Activities/ Media	Lecturer
		Lecture Hours	Lab/Field Trip/ Internship Hours		
1	Overview and basic elements of computer programs	3	2	Reading assignment, interactive lecture, quiz, group activities, case studies, group discussion	KLB SSC
2	Functions	3	2		
3-4	Conditional and iteration	6	4		
5-6	Functions as an abstraction; mixing control-flow constructs to write larger programs	6	4		
7-8	Files; Dictionary; nested lists; and Tuples	6	4		
9	Recursion	3	2		

Week	Topic	Number of Hours		Teaching Activities/ Media	Lecturer
		Lecture Hours	Lab/Field Trip/ Internship Hours		
10-11	Objected-oriented programming and advanced	6	4		
12	Transition to other languages and review	3	2		
	Total	36	24		

2. Plan for Assessing Course Learning Outcomes

2.1 Assessing and Evaluating Learning Achievement

a. Formative Assessment

- Worksheet
- Class discussion

b. Summative Assessment

(1) Tools and Percentage Weight in Assessment and Evaluation

Learning Outcomes	Assessment Methods	Assessment Ratio (Percentage)	
CLO 1 Recognize the role of algorithms in problem solving.	Homework & Quiz	5	10
	Examination	5	
CLO 2 Describe the outcome and behaviors of control-flow constructs in imperative programs written in Python or similar languages.	Homework & Quiz	8	18
	Examination	10	
CLO 3 Use and reason about data types, conditionals, iterations, functions, and recursion in the student's own code.	Homework & Quiz	8	18
	Examination	10	

Learning Outcomes	Assessment Methods	Assessment Ratio (Percentage)	
CLO 4 Select and suitably apply standard collections such as list, dictionary, etc.	Homework & Quiz	8	18
	Examination	10	
CLO 5 Explain the mechanics of classes and object-oriented programming and implement simple logic in object-oriented style.	Homework & Quiz	8	18
	Examination	10	
CLO 6 Write, test, and reason about small to medium-sized programs in Python	Homework & Quiz	8	18
	Examination	10	
			100

(2) Grading System

Grade	**Achievement**	**Final Score (% Range)**	**GPA**
A	Excellent	90-100	4.0
B+	Very good	85-89	3.5
B	Good	80-84	3.0
C+	Fairly good	75-79	2.5
C	Fair	70-74	2.0
D+	Poor	65-69	1.5
D	Very Poor	60-64	1.0
F	Fail	Less than 60	0.0

(3) Re-examination (If course lecturer allows to have re-examination)

N/A - (Not applicable with MUIC)

3. Student Appeals

N/A

Section 6 Teaching Materials and Resources

1. Textbooks and/or other documents/materials

- *Severance, Charles (April 9, 2016). Python for Everybody: Exploring Data in Python 3. CreateSpace Independent Publishing Platform. ISBN 978-1530051120.*
- Lecture notes provided by the lecturer

2. Recommended textbooks and/or other documents/materials

Selected readings from pertinent scientific journals and textbooks or video clips, as posted on the course's e-learning site

3. Other Resources (If any)

N/A

Section 7 Evaluation and Improvement of Course Management

1. Strategies for evaluating course effectiveness by students

   1.1 Student feedback of instructors, teaching methods and materials, and course content through MUIC student evaluation forms

2. Strategies for evaluating teaching methods

   2.1 Evaluation of effectiveness based on student evaluation scores and comments

   2.2 Evaluation through peer observations by co-instructor or other Division faculty

3. Improvement of teaching methods

   3.1 Adjustments based on student feedback, personal observations, comments from peer observations and discussions with supervisor and/or other Division faculty in one-on-one and/or group meetings as specified by MUIC guidelines

4. Verification process for evaluating students' standard achievement outcomes in the course

   4.1 Verification through student performance on assessments based on MUIC/Division standards

5. Review and plan for improving the effectiveness of the course

   5.1 Course instructors (and coordinator/supervisor) will meet to discuss results of student evaluations and student performance based on learning outcomes in order to identify point for improvement

   5.2 Strategy for improvement set according to MUIC/Division guidelines

Appendix
Alignment between Courses and General Education courses

Table 1 The relationship between course and Program Learning Outcomes (PLOs)

	Program Learning Outcomes (PLOs)					
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6
(ICCS101)	I	I			I	

Table 2 The relationship between CLOs and Program LOs (Number in table = Sub LOs)

ICCS101	Learning Outcomes in the Computer Science Program					
	1	2	3	4	5	6
CLO 1 Recognize the role of algorithms in problem solving.	1.2 1.3					
CLO 2 Describe the outcome and behaviors of control-flow constructs in imperative programs written in Python or similar languages.	1.2				5.1	
CLO 3 Use and reason about data types, conditionals, iterations, functions, and recursion in the student's own code.					5.1	
CLO 4 Select and suitably apply standard collections such as list, dictionary, etc.					5.1	
CLO 5 Explain the mechanics of classes and object-oriented programming and implement simple logic in object-oriented style.	1.2 1.3				5.1	
CLO 6 Write, test, and reason about small to medium-sized programs in Python		2.3			5.1	

Table 3 The description of Program LOs and Sub LOs of the course

CS LOs	Sub LOs
PLO1 Demonstrate proficiency in scientific communication.	1.1 Understand the format of communication in computer science.
	1.2 Communicate inchoate ideas to others for further development and refinement.
	1.3 Describe computing concepts to members of the community with accuracy and clarity.
PLO2 Carry out work with scientific integrity and professionalism.	2.1 Recognize the concepts of intellectual property, copyright licenses, and law pertaining to information technology.
	2.2 Provide ethical reasoning and awareness of issues surrounding bias, fabrication, falsification, plagiarism, outside interference, censorship, and information privacy.
	2.3 Demonstrate good time management, self-regulation, autonomy, and professional code of conduct of the discipline.
PLO3 Appraise scientific information critically.	3.1 Apply quantitative reasoning using mathematical methods and scientific facts, taking into consideration multiple perspectives.
	3.2 Provide a succinct description of the issue (i.e., a problem, a question, or a hypothesis), separating facts and assumptions.
	3.3 Differentiate source, validity, objectives, key arguments, and consequences of a piece information.
	3.4 Create a response to the issue by synthesizing collected information critical to the assessment.
PLO4 Use a teamwork mindset in the context of computing.	
PLO5 Execute common computing methodologies appropriate for a problem scenario.	5.1 Carry out the process of converting a process/ algorithm to a machine-executable program.
	5.2 Use suitable techniques for correctness and cost analysis of computer programs.

CS LOs	Sub LOs
	5.3 Deconstruct a computer system to reveal its structure, components, and process of construction.
	5.4 Select common computing techniques (e.g., standard algorithms, data structures, design patterns, programing style, and computing paradigms) appropriate for a given problem scenario.
PLO6 Formulate computational solutions to novel situations grounded on the foundation of computer science.	6.1 Model a given problem using suitable abstractions, including problem decomposition, in the context of computing.
	6.2 Compare the relative strengths and weaknesses among multiple designs or implementations.
	6.3 Assess the feasibility and efficacy of a computational solution based on its design and implementation.
	6.4 Devise computational solutions to novel situations using knowledge and experience in computer science.