

Annexe A: New/Revised Course Content in OBTL+ Format

Course Overview

Expected Implementation in Academic Year (New format)	AY2026-2027
Semester/Trimester/Others (specify approx. Start/End date)	Semester 2
Course Author * Faculty proposing/revising the course	Asst Prof Bryan German Pantoja Rosero
Course Author Email	bryan.pr@ntu.edu.sg
Course Title	Introduction to Computational Thinking
Course Code	CV1014
Academic Units	3
Contact Hours	39
Research Experience Components	Not Applicable

Course Requisites (if applicable)

Pre-requisites	
Co-requisites	
Pre-requisite to	
Mutually exclusive to	
Replacement course to	
Remarks (if any)	

Course Aims

Computational thinking (CT) is a problem solving process with the aid of computer; i.e. formulating a problem and expressing its solution in such a way that a computer can effectively carry it out. It includes a number of characteristics, such as breaking a problem into small and repetitive ordered steps, logically ordering and analyzing data and creating solutions that can be effectively implemented as algorithms running on computer. As such, computational thinking is essential not only to the Computer Science discipline, it can also be used to support problem solving across all disciplines, including math, science, engineering, business, finance and humanities.

The aim of this course is hence to take students with no prior experience of thinking in a computational manner to a point where you can derive simple algorithms and code the programs to solve some basic problems in your domain of studies. In addition, the course will introduce systematic testing and debugging practices, basic awareness of computational efficiency, and the responsible use of generative AI as a programming assistant, including the critical evaluation and validation of AI-generated outputs.

Course's Intended Learning Outcomes (ILOs)

Upon the successful completion of this course, you (student) would be able to:

ILO 1	Describe how a program is executed by a computer and computing trends.
ILO 2	Code basic programs based on the programming language used in the course
ILO 3	Formulate a problem and express its solution in such a way that a computer can effectively carry it out. (i.e. equip you with CT skills)
ILO 4	Apply the CT concepts on case studies/problem-based scenarios through hands-on practice of the CT processes.
ILO 5	Apply generative AI tools to support programming tasks while critically evaluating, validating, and responsibly using AI-generated outputs.

Course Content

1. Course Overview and Concepts of Computational Thinking
2. Problem Analysis and Algorithm Representation
3. Variables, Data Types, Expressions, and Sequential Programming
4. Boolean Logic and Selection Structures
5. Iteration and Repetitive Problem Solving
6. Functions and Procedural Decomposition
7. Strings, Lists, and Basic Data Collections
8. Pattern-Based Problem Solving, Searching, and Sorting
9. Testing, Debugging, and Efficiency Awareness
10. Introduction to AI-Assisted Programming (Vibe Coding)

Reading and References (if applicable)

The course will not use any specific text book. The following books and websites will be used as reference materials.

1. The Practice of Computing using Python; Enbody, Richard, and William Punch, Pearson, 2021.
2. Introduction to Computation and Programming Using Python : With Application to Understanding Data; (3rd Ed) John V. Guttag, MIT Press Ltd, 2021.
3. <https://edu.google.com/resources/programs/exploring-computational-thinking/>

The teaching faculty will furnish the updated reference if there are newer versions.

Planned Schedule

Week or Session	Topics or Themes	ILO	Delivery Mode	Activities	Readings
1	Introduction to Computational Problem Solving	1, 3, 4	In-person	Lecture and tutorial	Lecture notes
2	Representing Algorithms with Pseudocode and Flowcharts	3, 4	In-person	Lecture and tutorial	Lecture notes
3	Variables, Data Types, Expressions, and Sequential Programs	2, 3, 4	In-person	Lecture and tutorial	Lecture notes
4	Boolean Logic, Relational Operators, and Simple Selection	2, 3, 4	In-person	Lecture and tutorial	Lecture notes
5	Compound Selection, Logical Operators, and Multi-Case Decisions	2, 3, 4	In-person	Lecture and tutorial	Lecture notes
6	Iteration I: Definite Repetition, Counters, and Accumulators	2, 3, 4	In-person	Lecture; Written Quiz 1	Lecture notes
7	Iteration II: Indefinite Repetition, Sentinel Values, and Validation Loops	2, 3, 4	In-person	Lecture and tutorial	Lecture notes
8	Functions and Procedural Decomposition	2, 3, 4	In-person	Lecture and tutorial	Lecture notes
9	Strings, Lists, and Basic Data Collections	2, 3, 4	In-person	Lecture and tutorial	Lecture notes

Week or Session	Topics or Themes	ILO	Delivery Mode	Activities	Readings
10	Pattern-Based Problem Solving with Lists and Functions	2, 3, 4	In-person	Lecture and tutorial	Lecture notes
11	Introduction to AI-Assisted Programming (Vibe Coding)	2, 3, 4, 5	In-person	Lecture; Written Quiz 2	Lecture notes
12	Testing, Debugging, and Efficiency Awareness	2, 3, 4	In-person	Lecture and tutorial Mini project – coding and debugging	Lecture notes
13	Searching and Sorting as Reusable Algorithmic Strategies	2, 3, 4	In-person	Lecture and tutorial Mini project – presentation	Lecture notes

Learning and Teaching Approach

Approach	How does this approach support you in achieving the learning outcomes?
In-Person Lecture (2 hours / week)	The main theoretical and practical contents of the course will be delivered through weekly face-to-face lectures. The lectures will follow the course notes and introduce the relevant concepts, problem-solving methods, pseudocode, and Python programming structures. Short demonstrations, worked examples, and guided hands-on exercises will be integrated into the lectures to help students understand the theoretical concepts and connect them with their practical implementation. Reference reading materials will also be provided to support independent study.
Example Class (Face-to-Face, 1 hour)	The example class will provide additional guided practice to reinforce the concepts introduced during the lecture. Students will solve problems involving problem analysis, algorithm design, pseudocode, Python implementation, testing, and debugging. These sessions will also allow students to clarify questions, receive feedback, and progressively develop the practical skills required for the course exercises and mini-project.

Assessment Structure

Assessment Components (includes both continuous and summative assessment)

No.	Component	ILO	Related PLO or Accreditation	Weightage	Description of Assessment Component	Team/Individual	Rubrics	Level of Understanding
1	Continuous Assessment (CA): Assignment(CA1: Hands-on Exercises)	1, 2, 3, 4	a, b, c, d, e, f, g, h, i, j, k	10	Students will complete and submit selected exercises conducted during the tutorial sessions	Individual	Analytic	Multistructural
2	Continuous Assessment (CA): Test/Quiz(CA2: Quiz 1)	1, 2, 3	a, b, c, d, e, f, h, i, k	20	Written quiz conducted during tutorial sessions	Individual	Analytic	Multistructural
3	Continuous Assessment (CA): Test/Quiz(CA3: Quiz 2 - Written Quiz)	2, 3, 4	a, b, c, d, e, g, h, i, j, k	40	An individual written mandatory Quiz conducted outside the lecture and lab sessions	Individual	Analytic	Relational
4	Continuous Assessment (CA): Project(CA5: AI-Assisted Mini-Project)	2, 3, 4, 5	a, b, c, d, e, g, h, i, j, k	30	Mini-project report and presentation. Students work in groups of three to design and implement a working Python program as a mini-project. Generative AI may be used as a	Individual and Team	Holistic	Relational

No.	Component	ILO	Related PLO or Accreditation	Weightage	Description of Assessment Component	Team/Individual	Rubrics	Level of Understanding
					programming assistant. Students must document relevant AI use, critically evaluate generated suggestions, test and debug the final program, and remain responsible for the correctness and quality of the submitted solution. Individual contribution may be considered through peer assessment and an individual modification factor, as specified in the project rubric.			

Description of Assessment Components (if applicable)

** To pass this course, students are required to take this mandatory in-person Quiz 2, which is tentatively scheduled in Week 11.

Students who are unable to attempt the quiz due to valid reasons may sit for a make-up in-person quiz, tentatively scheduled on Week 13

Formative Feedback

For online tasks, immediately after you submitted the answers, you will see your scores, your answers, the correct answers, feedback on your incorrect answers, and explanations for the correct answers. For online and MCQ quizzes, individual feedback will be provided to students through evaluation of their submissions. Quiz answers will be discussed in the example class. you will also see the average scores of the other students in the same cohort.

For lab assessments, you will be given verbal feedbacks during your demonstrations of the circuits.

NTU Graduate Attributes/Competency Mapping

This course intends to develop the following graduate attributes and competencies (maximum 5 most relevant)

Attributes/Competency	Level
Collaboration	Intermediate
Creative Thinking	Intermediate
Problem Solving	Advanced
Transdisciplinarity	Advanced

Course Policy

Policy (Academic Integrity)

Good academic work depends on honesty and ethical behaviour. The quality of your work as a student relies on adhering to the principles of academic integrity and to the NTU Honour Code, a set of values shared by the whole university community. Truth, Trust and Justice are at the core of NTU's shared values. As a student, it is important that you recognize your responsibilities in understanding and applying the principles of academic integrity in all the work you do at NTU. Not knowing what is involved in maintaining academic integrity does not excuse academic dishonesty. You need to actively equip yourself with strategies to avoid all forms of academic dishonesty, including plagiarism, academic fraud, collusion and cheating. If you are uncertain of the definitions of any of these terms, you should go to the academic integrity website for more information. On the use of technological tools (such as Generative AI tools), different courses / assignments have different intended learning outcomes. Students should refer to the specific assignment instructions on their use and requirements and/or consult your instructors on how you can use these tools to help your learning. Consult your instructor(s) if you need any clarification about the requirements of academic integrity in the course.

Policy (General)

As a student of the course, you are required to abide by both the University Code of Conduct and the Student Code of Conduct. The Codes provide information on the responsibilities of all NTU students, as well as examples of misconduct and details about how students can report suspected misconduct. The university also has the Student Mental Health Policy. The Policy states the University's commitment to providing a supportive environment for the holistic development of students, including the improvement of mental health and wellbeing. These policies and codes concerning students can be found in the following link.
<http://www.ntu.edu.sg/SAO/Pages/Policies-concerning-students.aspx>

Policy (Absenteeism)

Each assessment is important for this course. Absence from any of the assessments without a valid reason will affect your overall course grade. Valid reasons include falling sick supported by a medical certificate and participation in NTU's approved activities supported by an excuse letter from the relevant bodies.

Also, to pass this course, students are required to take the mandatory in-person Quiz 2; students who are unable to attempt the quiz due to valid reasons may sit for a make-up in-person quiz.

Policy (Others, if applicable)

This course will adopt NTU's policy on the use of AI for take-home assignments and reports, and project reports and presentation slides. More details could be found via the link
[https://entuedu.sharepoint.com/sites/Student/dept/ctlp/SitePages/Exploring-the-Impact-of-Generative-Artificial-Intelligence-\(GAI\)-Tools-on-Education.aspx#1-proper-citations-and-acknowledgements](https://entuedu.sharepoint.com/sites/Student/dept/ctlp/SitePages/Exploring-the-Impact-of-Generative-Artificial-Intelligence-(GAI)-Tools-on-Education.aspx#1-proper-citations-and-acknowledgements)

Technical Skills To Be Developed

All skills to be developed for this course:

Computational Design
Computational Modelling
Engineering Problem Solving
Generative AI Prompt Engineering
Mathematical Concepts Application
Programming and Coding
Responsible AI and Generative AI Practices
Solutioning

Last Updated Date: 28-08-2026 12:25:29

Last Updated By: Qian, Shunzhi