COURSE CONTENT

Academic Year	AY19-20		**Semester**		2	
Course Coordinator	Zhang Limao (limao.zhang@ntu.edu.sg)					
Course Code	CV1014					
Course Title	Introduction to Computational Thinking					
Pre-requisites	NIL					
No of AUs	3					
Contact Hours	Lectures	0	TEL(Online Videos)	13	Example Class (Seminars/Hands-on Exercises)	26
Proposal Date	November 2019					

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 include topics to appreciate the internal operations of a processor, and raise awareness of the socio-ethical issues arising from the pervasiveness of computing technology.

Intended Learning Outcomes (ILO)

Upon the successful completion of this course, you shall be able to:

1. Describe the internal operation of a basic processor, how a program is executed by a computer and computing trends.
2. Code basic programs based on the programming language used in the course
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)
4. Apply the CT concepts on case studies/problem-based scenarios through hands-on practice of the CT processes.

Course Contents

No	Topics	LAMS (Hours)	Example Class (2-hr session)
1	**Course Overview and Concepts of Computational Thinking** Solving complex problem using computer **-** enables the student to work out exactly what to tell the computer to do.	0.5	Programming Languages Exercises (1 week)
2	**Overview of Programming Languages** Graphic programming, high level programming languages (Python, C, Java, R, Matlab)	0.5	
3	**Basic internal operation of computer** Basic computer organization (Processor, Memory, I/O) and how a computer execute a program (Machine instructions)	1	Computer Operations Exercises (1 week)
4	**Basic program structure: control constructs and data types** Concepts of data types, variables; Pseude code and flowcharts; Sequences, Selection (if/else), iteration (for/while loop);	3	Python Programming Exercises/Quiz (3 weeks)
5	**CT concept - Abstraction** Problem formulation - reducing something to a very simple set of characteristics to only focusing on the most relevant to the problem. Concept of functions/libraries and data structure	2	Library calls & function (2 weeks)
6	**CT concept - Decomposition** Break a complex problem into smaller and more manageable parts/steps, such that each of these smaller problems can then be looked at individually	1	Project – Flow Chart Design (1 week)
7	**CT concept – Pattern recognition** Looking for similarities among and within problems, which also enable re-use knowledge of previous similar problems	1	Project – Coding (1 week)
8	**CT concept – Algorithm** Reformulating the problem into series of ordered steps through Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources. (Some common/useful examples: Sorting and searching)	2	Project – Coding/Quiz (2 weeks)
9	**Limit of computing** Analysis of Algorithm Complexity to determine how much resources (space and time) are needed to execute an Algorithm in order to achieve code optimization.	1	Project – Demo and assessment (1 week)
10	**Computing Trends** Cloud, Edge and Fog computing, Quantum Computers	0.5	Project – Demo and assessment

11	**Social-Ethical Issues and Ramifications of Computing** Fairness, Privacy, Sharing, Hacking, software Piracy, Data Protection, Cyberbullying and trolling, Fake news, digital divides, IP/Copyright	0.5	(1 Week)
	Total Hours	13	26

Assessment (includes both continuous and summative assessment) CV1014

Component	Course LO Tested	Related Programme LO or Graduate Attributes	Weightage	Team / Individual	Assessment Rubrics
1. TEL and Hand-on exercise participations	1,2,3,4,	a, b, c, d	10%	Individual	See appendix 1
2. TEL MCQs	2,3,4,	a, b, c	20%	Individual	See appendix 2
3. Hands-on exercises completion and assessment	2,3,4,	a, b, c	40%	Individual	See appendix 3
4. Mini Project assessment	2,3,4,	a, b, c, d, e, i, j, l	30%	Group and Individual	See appendix 4
Total			100%		

Mapping of Course SLOs to EAB Graduate Attributes

Course Student Learning Outcomes	Cat	EAB's 12 Graduate Attributes*												EAB's CE/CS requirements	
		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	CE	CS
CV1014 Digital Logic	Core	●	●	●	●	◐				◐	O		◐	●	●
Overall Statement	This course aims to take students with no prior experience of thinking in a computational manner to a point where they can derive simple algorithms and code the programs to solve some basic problems in their domain of studies. In addition, the course will include topics to appreciate the internal operations of a processor, and raise awareness of the socio-ethical issues arising from the pervasiveness of computing technology													Apply to both CE and CS programmes	
1. Describe the internal operation of a basic processor, how a program is executed by a computer and computing trends.	a, b, c, d														
2. Code basic programs based on the programming language used in the course	a, b, c, d, e, i														
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)	a, b, c, d, e, i, j, l														
4. Apply the CT concepts on case studies/problem-based scenarios.	a, b, c, d, e, l														

Legend:

● Fully consistent (contributes to more than 75% of Student Learning Outcomes)

◐ Partially consistent (contributes to about 50% of Student Learning Outcomes)

O Weakly consistent (contributes to about 25% of Student Learning Outcomes)

Blank Not related to Student Learning Outcomes

*The graduate attributes as stipulated by the EAB, are:

(a) **Engineering knowledge**: Apply the knowledge of mathematics, natural science, engineering fundamentals, and an engineering specialisation to the solution of complex engineering problems.
(b) **Problem Analysis**: Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
(c) **Design/development of Solution**s: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
(d) **Investigation**: Conduct investigations of complex problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
(e) **Modern Tool Usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
(f) **The engineer and Society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
(g) **Environment and Sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for the sustainable development.
(h) **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
(i) **Individual and Team Work**: Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.
(j) **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
(k) **Project Management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and economic decision-making, and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
(l) **Life-long Learning**: Recognise the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

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.

Learning and Teaching approach

Approach	How does this approach support students in achieving the learning outcomes?
TEL (Online Video)	Topics will be delivered as a series of online videos lectures, and you will also be provided reference reading materials for self-study to achieve LO 1 to LO 4
Example Class (Face-to-face)	The Example class will be used as seminar sessions for students to clarify the contents of the online topic, as well as hands-on sessions to equip you with practical knowledge on coding, and on the design and implementation of a mini project to achieve LO 1 to LO 4.

Reading and References

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; William Punch and Richard Enbody, Pearson, 2017.
2. Introduction to Computation and Programming Using Python : With Application to Understanding Data; (2nd Ed) John V. Guttag, MIT Press Ltd, 2016.
3. https://edu.google.com/resources/programs/exploring-computational-thinking/

Course Policies and Student Responsibilities

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

Academic Integrity

Good academic work depends on honesty and ethical behavior. The quality of your work as a student relies on adhering to the principles of academic integrity and to the NTU Honor 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, and 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. Consult your instructor(s) if you need any clarification about the requirements of academic integrity in the course.

Course Instructors

Instructor	Office Location	Phone	Email
Zhang Limao	N1-01b-42	6790 5272	limao.zhang@ntu.edu.sg

Planned Weekly Schedule

Week	Topic	Course LO	Readings	Example Class Activities
1	Computational Thinking Concepts, Programming languages	3,4	On-line Video	Programming Languages exercises
2	Basic internal operation of computer	1	On-line Video.	Computer operations exercises
3	Basic program structure: Case Study, Pseudo code and flowchart,	2	On-line Video.	Python programming exercises
4	Basic program structure: Data type, Variable, sequence, logic and comparison operation	2	On-line Video.	Python programming exercises
5	Basic program structure: Selection and Iteration	2	On-line Video	Python programming exercises
6	Procedural abstraction: function and library	2,3	On-line Video	Python programming exercises (Function)
7	Data abstraction: Data structure	2,3	On-line Video.	Python programming exercises (Function/Data Structures)
8	Decomposition Case study	2,3,4	On-line Video	Mini project – Flow Chart Design
9	Pattern recognition Case study	2,3,4	On-line Video	Mini project – Coding
10	Algorithms	2,3,4	On-line	Mini project –

	Sorting algorithm		Video	Coding and debugging
11	Algorithm design Searching algorithm	2,3,4	On-line Video	Mini project – Coding and debugging
12	Algorithm Complexity Analysis Big-0 concept	2,3,4	On-line Video	Mini project – Testing and assessment
13	Computing trends and Ethical considerations	1	On-line Video	Mini project – assessment

Appendix 1: Assessment on TEL and Hand-on exercise participations
You will do 13 lab exercises and complete 12 online videos, and will be assessed according to your participation. The maximum score is 10.

Appendix 2: Assessment on TEL MCQs
You will need to complete 12 online videos with MCQs based quizzes. The maximum score is 20.

Appendix 3: Assessment Criteria for Hands-on programming exercise
You will take 2 MCQs based quizzes related to the programming exercises done in the class. The maximum score is 40.

Appendix 4: Assessment Criteria for Mini Project
You will demonstrate 1 working program in the form of a mini project. The maximum score is 30.

Criteria	**Standards**		
	Fail standard (0-39%)	**Pass standard (40-80 %)**	**High standard (81-100 %)**
Demonstrate (including explanation) the use of CT concepts in the implementation of the project. (LO 2,3,4)	Demonstrated less than 40% of the functionalities according to the specifications.	Demonstrated 40% to 80% of the functionalities according to the specifications.	Demonstrated more than 80% of the functionalities according to the specifications.
Individual assessment (LO 2,3,4)	Unable to explain individual work performed. Unable to understand and answer questions.	Reasonable depth of explanation on individual work performed. Able to understand and answer some questions.	Offer good explanation on individual work performed. Able to understand and answer most questions.