CIEE Global Institute – Copenhagen

Course name:	Computer Science I Introduction to Programming
Course number:	(GI) CMPS 1001 CPDK
Programs offering course:	Copenhagen Open Campus Block
Open Campus track:	STEM and Society
Language of instruction:	English
U.S. semester credits:	3
Contact hours:	45
Term:	Spring 2020

Course Description

This course is an introduction to computer science based on computational thinking, application, problem-solving and basic programming. It explores what computers do and how you can make them do it. It covers fundamental concepts like basic data types and their operators, control structures, classes and objects, files, sets, arrays, dictionaries and searching algorithms. Students will gain fundamental knowledge of computer hardware, software, programming languages, and how to use basic features of standard productivity software. The course explores programming, debugging and problem-solving using several different computer languages, but mostly Python. Applications include web design (HTML), image processing, file transfer protocols, with an introduction to numerical computing, graphics and artificial intelligence concepts.

Learning Objectives

Upon completion, students taking this course will be able to:

- Describe the purpose and behavior of basic computer programming.
- Design algorithms and write small-scale computer programs with real world applications.
- Apply multiple applications of computational thinking to real-world problems both orally and in writing.
- Understand and articulate ethical and professional issues in computer science and the impact of computers on culture and society, both orally and in writing.
- Apply basic object-oriented analysis and design.
- Develop programs by applying variables, basic and composite data and collections.
- Construct programs using sequence, selection and iteration structures, primarily using Python.
- Articulate how Computer Science impacts their life, as well as the local culture, regional

identity, and function of our global society, both orally and in writing.

Course Prerequisites

None

Methods of Instruction

Students will attend interactive lectures that lead to problem-solving workshops, discussions and interpretation. Instruction emphasizes experiential learning, participation and applications. Students will use critical thinking to connect recent literature, historical perspectives, experimental findings, theory and quantitative analysis. Discussions and written assignments develop a better understanding of Psychology, as well as how psychological science informs culture and society. Excursions will showcase local Computer Science application and innovation.

Assessment and Grading

Participation	20%
Weekly Quizzes (6)	15%
Activity Worksheets (5)	15%
Program 1	20%
Computers and Culture Essays (2)	10%
Program 2	20%
Total	100%

Course Requirements

Weekly Quizzes

Each week, students will take a quiz on the previous week's course material, including lectures, activities and readings. Quizzes will have True/False, Multiple Choice, filling in blanks and short answer questions. Quizzes will cover only new material from that week but will build on previous concepts.

Activity Worksheets

During and after lectures, students will have a series of tasks, discussions and demonstrations related to the lecture material. They will work both individually and in groups to complete the tasks, handing in answers to a series of questions, working with simple code or debugging simple program script before leaving the class.

Program 1

Small groups of students will work with instructors to work with code and write a simple program of at least 20 lines that completes a novel action. This program should address a real-world issue that has some cultural or social impact.

Computer Science and Culture Essays

Students will use online resources to explore the intersection between computer science and culture. Students will write two 500-word essays: one on a cultural feature of the host culture and another comparing two cultures. In each case, Computer Science concepts will be used in the analysis and to make major points.

Program 2

Individual students will work with instructors to take a problem and use computer science to design an appropriate, object-oriented program with at least 100 lines. This program will be written in Python and address a real-world problem with some cultural or social impact.

Participation

Participation is valued as meaningful contribution in the digital and tangible classroom, utilizing the resources and materials presented to students as part of the course. Meaningful contribution requires students to be prepared in advance of each class session and to have regular attendance. Students must clearly demonstrate they have engaged with the materials as directed, for example, through classroom discussions, online discussion boards, peer-to- peer feedback (after presentations), interaction with guest speakers, and attentiveness on co-curricular and outside-of-classroom activities.

Class Attendance

Regular class attendance is required throughout the program, and all unexcused absences will result in a lower participation grade for any affected CIEE course. Due to the intensive schedules for Open Campus programs, unexcused absences that constitute more than 10% of the total course will result in a written warning.

Students who transfer from one CIEE class to another during the add/drop period will not be considered absent from the first session(s) of their new class, provided they were marked present for the first session(s) of their original class. Otherwise, the absence(s) from the original class carry over to the new class and count against the grade in that class.

For CIEE classes, excessively tardy (over 15 minutes late) students must be marked absent. Attendance policies also apply to any required co-curricular class excursion or event, as well as to Internship, Service Learning, or required field placement. Students who miss class for personal travel, including unforeseen delays that arise as a result of personal travel, will be marked as absent and unexcused. No make-up or re-sit opportunity will be provided.

Attendance policies also apply to any required class excursion, with the exception that some class excursions cannot accommodate any tardiness, and students risk being marked as absent if they fail to be present at the appointed time.

Unexcused absences will lead to the following penalties:

Percentage of Total Course Hours Missed	*Equivalent Number of Open Campus Semester classes*	*Minimum Penalty*
Up to 10%	1 content classes, or up to 2 language classes	Participation graded as per class requirements
10 – 20%	2 content classes, or 3-4 language classes	Participation graded as per class requirements; **written warning**
More than 20%	3 content classes, or 5 language classes	Automatic **course failure**, and possible expulsion

Weekly Schedule

NOTE: this schedule is subject to change at the discretion of the instructor to take advantage of current experiential learning opportunities.

Week 1 Introduction to Computer Science and Python

Session 1.1 Computing

Students explore what Computer Science encompasses and discuss how it potentially interacts with culture and society. They investigate how to measure computer power, exploring bits, bytes and binary information along the way. They apply this to sequences of bits and how they can describe different types of data, like numbers, discrete values, text, images, audio and movies. They consider the tremendous growth in computing power and how this allows innovation. They discuss how Computer Science is an amalgam of Science, Engineering and the Liberal Arts. Students will work in small groups to explore computer components and computer design. They will learn to use an abacus, astrolabe, slide rule and examine cuckoo clocks, applying their design principals to modern day computers.

Readings: Chapter 1 Getting Started and Turkle, S. 2004. How Computers Change the Way We Think. The Chronicle of Higher Education 50: B26 found at http://web.mit.edu/sturkle/www/pdfsforstwebpage/Turkle_how_computers_change_way_we_think.pdf

Watch: Early Computing –Crash Course Computer Science #1. 2017. https://www.youtube.com/watch?v=O5nskjZ_GoI

Week 2 Numerical Programs

Session 2.1 Working in Python

Students introduce themselves to Python, reviewing its fundamental elements. They define Objects, Expressions and Numerical Types. They describe Variables and Assignments in the context of Python. They work with Branching Programs, Strings and Input, as well as Iteration. Students will work in small groups with simple programs to begin building with and deconstructing the basic logic of computer programs using Python.

Readings: Chapter 2 Introduction to Python and Zobel, J. 2016. The Computer Revolution: How It's Changed Our World Over 60 Years. World Economic Forum at https://www.weforum.org/agenda/2016/06/the-computer-revolution-how-its-changed-our-world-over-60-years

Watch: Aaron, S. 2015. Programming as Performance. TEDx Talks. https://www.youtube.com/watch?v=TK1mBqKvIyU

Session 2.2 Simple Numerical Programs

Students explore Exhaustive Enumeration. They then move onto consider Loops, Approximate Solutions and Bisection Search. They discuss Floats and investigate the Newton-Raphson as a root-finding algorithm for finding better approximations to the roots or zeroes of a real-valued function. Students work in small groups to apply these concepts using Python.

Reading: Chapter 3 Some Simple Numerical Programs

Watch: Liukas, L. 2015. The Poetry of Programming. TEDx Talks. https://www.youtube.com/watch?v=-jRREn6ifEQ and and Boolean Logic & Logic Early Programming –Crash Course Computer Science #10. 2017. https://www.youtube.com/watch?v=nwDq4adJwzM&t=45s

Quiz 1 (covers material from Week 1)

Watch: Tasman, M. 2015. Future Cultures. TEDx Talk. https://www.youtube.com/watch?v=WpJmqC1gTP4

Due: Activity Worksheet

Week 3 Functions

Session 3.1 Functions, Scoping and Abstraction

Students gain greater appreciation that programming is writing, and that it is a creative process to make the computer do what you want. They see that good programs are written to be efficient, but also to be read and understood by people. To do this, students apply concepts of Function and Scoping, including function definitions, keyword arguments and default values, and scoping. Students then explore recursive applications, applying them to Fibonacci

Numbers and palindromes. They also work with global variables, modules and files.

Quiz 2 (covers material from Week 2)

Watch: The First Programming Languages – Crash Course Computer Science #11 https://www.youtube.com/watch?v=RU1u-js7db8 and Programming Basics: Statements and Functions – Crash Course Computer Science #12 https://www.youtube.com/watch?v=l26oaHV7D40

Readings: Chapter 4 Functions, Scoping, and Abstraction and Memmi, D. 2013. Cultural Consequences of Computing Technology. AI and Society 28: 77-85.

Session 3.2 Structured Types, Mutability and Higher-Order Functions

Students will define and work with tuples as a type of sequence. They will differentiate tuples from other sequence types, like lists and examine Tuple unpacking or multiple assignment use in Python. They will use the Range function to generate lists of numbers and for loops. They also apply lists and mutability to cloning and list comprehension. Students then explore Functions as Objects and Dictionaries. Students will work in groups to put these concepts into practice. They will also begin work to construct Program 1.

Readings: Chapter 5 Structured Types, Mutability, and Higher-Order Functions

Optional Watch: A Beautiful Mind

Due: Activity Worksheet, Computer Science and Culture Essay 1

Session 3.3 Programming Workshop

Students apply concepts studied this week. Then, they will work in groups on Program 1, a program written to solve a tangible program that benefits culture or society in some way.

Reading: Tsekeris, T., Tsekeris, C. and Katerelos, I., 2018. Reflections on networks, human behavior, and social dynamics in the digital age. *AI & SOCIETY*, *33*(2), pp.253-260.

Watch: Intro to Algorithms – Crash Course Computer Science #13. https://www.youtube.com/watch?v=rL8X2mlNHPM

Due: Activity Worksheet

Week 4 Sorting, Searching and Improving Expressiveness

Session 4.1 Testing and Debugging, Exceptions and Assertions

In this session, students use Python in testing programs, including Black-Box and Glass-box Testing. They learn to debug programs, as well. They consider them importance of experiment design in debugging, what to do when the going gets tough, and what do to if they find a bug. Students will work in groups to debug model programs. They also define and work with exceptions and assertions.

Quiz 3 (covers material from Week 3)

Readings: Chapter 6 Testing and Debugging and Chapter 7 Exceptions and Assertions

Due: Program 1

Session 4.2 Visit to Computer Science Leading Institute.

During this visit, students will speak with professional Computer Science leaders about current projects connecting computer science to culture and society. Students will have a tour, speak with several researchers and discuss the impact of computers on culture and society. Students will then use online resources to further explore two cultures, explaining similarities and differences in their reaction to and impact by computer science.

Readings: Has the technology revolution changed culture?
Debate.org https://www.debate.org/opinions/has-the-technology-revolution-changed-culture

Session 4.3 Programming Workshop

After applying this week's concepts in a variety of activities, students will work in groups to write and debug simple programs used to calculate outcomes for simple games.

Reading: Chapter 9 Mutation

Due: Activity Worksheet, Computer Science and Culture Essay 2

Week 5 Object-Oriented Programming

Session 5.1 Classes and Object-Oriented Programming

Students define and differentiate data types and classes. They investigate programs designed using abstract data types and how classes can be used to keep track of students and faculty. From there, the move on to the concept of inheritance in computer programming. They explore multiple levels of inheritance. They consider encapsulation and information hiding, including generators. They use mortgages as one application and explore others.

Quiz 4 (covers material from Week 4)

Readings: Chapter 8 Classes and Object-Oriented Programming

Watch: Data Structures – Crash Course Computing #14.
https://www.youtube.com/watch?v=DuDz6B4cqVc&t=34s

Session 5.2 Algorithm Complexity

Students consider computational complexity, classifying computational problems according to their inherent difficulty, including the use of asymptotic notation. They explore different complexity classes, including constant, logarithmic, loglinear, polynomial and exponential. They compare complexity classes and work on methods to assess complexity. They discuss practical limits on what computers can do.

Readings: Chapter 9 A Simplistic Introduction to Algorithmic Complexity

Watch: Partovi, H. 2014. Computer Science is for Everyone. TEDx.
https://www.youtube.com/watch?v=FpMNs7H24X0

Session 5.3 Programming Workshop

Students use algorithms to improve and extend several programs using Python. They will also begin work on Program 2, an individual program of at least 100 lines that addresses some tangible need.

Readings: Chapters 12 Data Analysis and Interpretation 2: Tests of Statistical Significance

Due: Activity Worksheet

Week 6 Algorithms and Data Structures

Session 6.1 Algorithms, Data Structures

Students work with search algorithms. They examine linear search and using indirection to access elements. They also investigate binary search and exploiting assumptions. They explore sorting algorithms, merge sort and exploiting functions as parameters. Students see how sorting works in Python. Finally, they implement and interpret simple hash tables.

Quiz 5 (covers material from Week 5)

Readings: Chapter 10 Some Simple Algorithms and Data Structures

Session 6.2 Programming Workshop

Students will review key concepts by writing several programs using key elements covered in the course. Students complete Program 2.

Session 6.3 Independent Programming Presentations

Students will present their findings to one another in a 5-8-minute PowerPoint presentation. They will demonstrate that their individual programs work successfully and explain how they address an actual social or cultural problem.

Final Quiz 6 (covers material from Week 6)

Due: Program 2

Course Materials

Course Textbook

Guttag, J., 2016. *Introduction to Computation and Programming Using Python: with Application to Understanding Data*. MIT Press.

Readings

Memmi, D. 2013. Cultural Consequences of Computing Technology. AI and Society 28: 77-85.

Tsekeris, T., Tsekeris, C. and Katerelos, I., 2018. Reflections on networks, human behavior, and social dynamics in the digital age. *AI & SOCIETY*, *33*(2), pp.253-260

Turkle, S. 2004. How Computers Change the Way We Think. The Chronicle of Higher Education 50: B26.

Zobel, J. 2016. The Computer Revolution: How It’s Changed Our World Over 60 Years. World Economic Forum

Online Resources

Python for Beginners https://www.python.org/about/gettingstarted/