CSCI-1010.M01:
Introduction to Computer Science

Credits: 3
Semester: Fall, 2019
Times: MWF 12:30-1:15
Prerequisite: MATH-120 or 3 years of high school math
Curriculum: Core science requirement, computer science

I. Course Description and Objectives

A broad survey of the computer science discipline, focusing on the computer's role in representing, storing, manipulating, organizing, and communicating information. Topics include hardware, software, algorithms, operating systems, networks.
The main objective of this course is to provide a basic understanding of the role, activities (design, analysis, and implementation [programming] of algorithms) by the computer scientist in the modern computer. The interface of hardware, software, operating systems, and programs will be the center of study.

II. Learning Outcomes:

The student upon completion of this course will be able to:

- Explain the organization of the classical von Neumann and Turing machines and their major functional units, including logic circuits and the CPU.
- Understand binary data representation in the modern computer, including the representation of non-numeric data, and standardized file structures. Understand how fixed-length number representations affect accuracy and precision in computing.
- Identify the necessary properties of good algorithms. Discuss the importance of algorithms in the problem solving process. Understand the software development cycle, good coding style, and algorithm development.
- Introduce the syntax of the Python programming language and develop small algorithmic programs in Python.
- Knowledge of typical algorithms used by computer scientists (for example, searching, sorting, and data compression) and O ( ) factors.
- Evaluate the choice of Abstract Data Structures – Queues, Lists, Arrays, etc. and their Python-specific implementation.

III. Textbooks and Readings

Recommended: For a centralized location of introductory topic material, the textbook available in the library:
Computer Science Illuminated, by Nell Dale and John Lewis, 5th Edition.

IV. Instructor Information

Name	Anas Abbood
E-mail:	anas.abbood@slu.edu
Office Hours	Tuesday & Thursday 2:00 – 3:30
Biography	PhD: Computer Science and Information Technology (Computer Graphics) MSc: Computer Science and Information Technology (Information Security) High Diploma: Computer Science and Information Technology (Information Security) Degree: Operation Research Area of Interest: Human-Computer interaction, Vision and Visualization, Computer Integrated System in Health, Geometric Modelling and Augmented Reality.

V. Course Outline (5 sections)

i. Introduction to Computing

Week 1: Welcome to the Digital Age, Number Systems: Base Arithmetic, Computer Mathematics, Artificial Intelligence.

Week 2: Data**,** Information, and It's Computer Representation/ **Project #1**

Week 3: Working with Computer Information and Physical Implementation of Data and Information in Computers.

Week 4: Computer Hardware and Architecture.

Week 5: Operating Systems; What is an Operating System? Parts of an Operating Systems; Basic Linux and Windows Commands/ **Project #2**

Week 6: Introduction to Networking.

ii. -Introduction to Computer Programming

Week 7: Universal Abstract Data Types/ **Midterm Exam**

Week 8: Introduction to Python Programming Language.

Week 9: Problem Solving and Algorithm Implementation in Python.

iii. Understanding, Encryption Algorithms and File and Network Security

Week 10: Data Representation - Binary Trees /Huffman Coding/

Week 11: Advanced Encryption Algorithms/**Project #3**

iv. Introduction to Simulation and Modelling Languages

Week 12: Using Excel in Engineering Simulations

Week 13: Advanced Engineering Simulations/**Project #4**

v. Android Applications and Programming

Week 14: Using GPS Location Devices and Implementing an Android App in AppInventor2/**Project #5**

Week 15: Using GPS Location Devices and Implementing an Android App in AppInventor2/**Final Exam**

VI. Grading System

The grade will be obtained from the following areas:

Homework (6):	30%
Projects (5):	30%
First Mid-Term Exam:	20%
Final Exam:	20%

Both Homework and Project assignments are made to be completed by the next week.

Any late work will be penalized at 5% per day late. If for example homework was due on Tuesday, but not submitted until Thursday the maximum grade for a 100% correct paper will be 90%.

Grading scale
100 < A < 95%,
95% <A- < 90%
90% < B+ < 87%
87% < B < 83%
83% < B-< 80%
80% < C+ < 77%
77% < C < 73%
73% < C- < 70%
70% < D < 50%
F < 50%

IX. Specific Course Policies

(1) Students are encouraged to participate in class discussions and to ask questions.
(2) Announcements may be made during the semester which alter the course content.
(3) Syllabus, reading and homework problems are subject to change.
(4) Students are responsible for all lecture material, handouts, homework and assigned reading.
(5) It is mandatory to attend all classes unless a reasonable excuse is given.
(6) Make up exams are not given. Students who legitimately miss an exam, due to a doctor's visit or family emergency must provide written documentation of the circumstances. A letter from the university counselor is accepted. Exams that are missed illegitimately result in a score of F. Grades for these students will be based on the remaining exams. Missing more than one exam always results in an F grade.

Detailed Course Outline

Day	Date	Topic	Web Sites/Readings
1	1/9	**Introduction to Number Systems/Artificial Intelligence** • Introduction to Digital Age/Information • Positional notation • Finite number sequences – single binary digit (bit), four binary digits (nibble), eight binary digits (Byte) • Use of different number systems – specifically base 2, Base 8, Base 16 with reference to Base 10 • Addition, Subtraction, Multiplication, Division in ALL Bases • Fractions with Bases • Finite representation errors generated by Base Conversion	http://courses.cs.vt.edu/~csonline/NumberSystems/Lessons/HexAndOctalNumbers/index.html http://www.howstuffworks.com/mp3.htm http://courses.cs.vt.edu/~csonline/NumberSystems/Lessons/index.html
2	3/9	**Introduction to Number Systems** • Integer Representation • Floating-point Representation (Decimal point = radix point)	**HOMEWORK #1 ASSIGNED** http://courses.cs.vt.edu/~csonline/NumberSystems/Lessons/index.html

		• Multiplication, Division in ALL Bases • Problems with negative numbers • One's Complement Math • Two's Complement Math	
3	8/9	**Understanding What Constitutes Information** • Introduction to Information • Representing Text • ASCII Character Set • Unicode Character Set • Text Compression • Keyword encoding • Run-length encoding (extension to graphic files - JPG)	Chapter 7 of *The Information: History, Theory, Flood* https://users.dcc.uchile.cl/~hsarmien/libros/The_Information_%20A_History.pdf http://www.fieggen.com/ian/g_formats.htm
4	10/9	**Understanding What Constitutes Information** • Huffman Coding Schemes (variable length binary encoding) • Audio Formats – WAV, AU, AIFF, VQF, MP3, MP4 • Images and Graphics - JPG,GIF, TIFF, PNG • Representing Color RGB, CYMK • Vector Images - Illustrator, Flash • Video Formats – CODECS (Compressor/Decompression) Real Video or MPEG2, MPEG4	**PROJECT #1 ASSIGNED** http://sinus.if.pw.edu.pl/podziemski/wp content/uploads/downloads/2012/05/jpegtool.pdf http://www msp.ece.mcgill.ca/documents/audioformats/index.html https://ccrma.stanford.edu/courses/
5	15/9	**Translating Information to the Physical Computer** • Introduction to Gates (Only 2 inputs) • Half-Adder Circuit • Full-Adder Circuit • Memory Circuits • CPU and GPU Chips	http://www.ee.surrey.ac.uk/Projects/Labview/gatesfunc/index.html http://inst.eecs.berkeley.edu/~ee42/sp04/lectures/lecture14student.pdf

6	17/9	**Translating Information to the Physical Computer** • Using Circuit Design Software	**FIRST MIDTERM ASSIGNED** http://www.ee.surrey.ac.uk/Projects/Labview/gatesfunc/index.html http://home.anadolu.edu.tr/~atdogan/EEM232/06-CombCktDesign.ppt
7	22/9	**Understanding Computer Architecture and Components** • CPU • Memory • BIOS • Motherboard • Ports and Slots	http://mathworld.wolfram.com/TuringMachine.html
8	24/9	**Understanding Computer Architecture and Components** • How the CPU Works • Machine Language • Assembler Language • High-Level Languages – Interpreted or Compiled • Super Simple CPU Software • Implementing Simple Algorithms (GCM, LCM)	**PROJECT #2 ASSIGNED** http://www.youtube.com/watch?v=cNN_tTXABUA
9	29/9	**Introduction to Operating Systems**	http://courses.cs.vt.edu/~csonline/OS/Lessons/Introduction/index.html http://ocw.uc3m.es/ingenieria-informatica/operating-systems/lecture-notes-1/Mt_t1_L1.pdf
10	1/10	**Introduction to Operating Systems**	http://courses.cs.vt.edu/~csonline/OS/Lessons/Introduction/index.html

			http://www.cmpe.boun.edu.tr/~uskudarli/courses/cmpe235/os.pdf
11	6/10	**Introduction to Networking** Hardware	
12	8/10	**Introduction to Networking** Software and Protocols	
13	13/10	**Understanding Abstract Data Types** • Queues • Arrays • Lists • Linked Lists • Trees	**HOMEWORK #3 ASSIGNED** http://www.idevelopment.info/data/Programming/data_structures/overview/Data_Structures_Algorithms_Introduction.shtml http://cslibrary.stanford.edu/103/LinkedListBasics.pdf
14	15/10	**Understanding Abstract Data Types Functions** • Sorting • Filtering	http://ww3.algorithmdesign.net/handouts/MergeSort.pdf http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=sorting
15	20/10	**Programming in Python** • Introduction to the General Structure of Python • General Data Types in Python • Using the Python Script and Interactive Modes	**Thinking as a Computer Scientist in Python** http://tdc-www.harvard.edu/Python.pdf https://www.softwaretestinghelp.com/python/python-data-types/ https://en.wikibooks.org/wiki/Python_Programming/Interactive_mode
16	22/10	**Programming in Python** • Implementing Abstract Data Types in Python – Dictionaries, Lists, Tuples • Control Statements in Python	**HOMEWORK #4 ASSIGNED** **Thinking as a Computer Scientist in Python** https://docs.python.org/3/library/datatypes.html http://openbookproject.net/thinkcs/python/english3e/dictionaries.html https://docs.python.org/3/library/collections.html#namedtuple-factory-function-for-tuples-with-named-fields
17	27/10	**Problem Solving, Algorithms, and Programming in Python**	**Thinking as a Computer Scientist in Python** **PROJECT #3 ASSIGNED** http://www.huffmancoding.com/david/algorithm.html

			https://www.cs.auckland.ac.nz/compsci105s1c/resources/ProblemSolvingwithAlgorithmsandDataStructures.pdf
18	29/10	**Problem Solving, Algorithms, and Programming in Python**	
19	3/11	**Binary and Other Trees** In-class Project: Implementing Huffman Coding in Python	
20	5/11	**Binary and Other Trees** In-class Project: Implementing Huffman Coding in Python	http://bhrigu.me/blog/2017/01/17/huffman-coding-python-implementation/
21	10/11	**Understanding Today's Encryption Algorithms and Protecting File Data** • Asymmetric Keys (Public/Private) • RSA Algorithm	**PROJECT #3 ASSIGNED** http://www.esat.kuleuven.be/cosic/intro/ https://www.esat.kuleuven.be/cosic/introducing-hector-project-first-year-results/
22	12/11	**Understanding Today's Encryption Algorithms and Protecting File Data** • Asymmetric Keys (Public/Private) • RSA Algorithm	http://www.esat.kuleuven.be/cosic/intro/
23	17/11	**Introduction to Simulation and Modeling Languages for Computer Scientists and Engineers** • Model Structures in Excel • Introduction to Excel • AutoFill • Formulas • Functions	http://oit.wvu.edu/training/files/excel2010_intro.pdf http://bookboon.com/en/excel-2010-introduction-part-i-ebook https://codecn.top/curriculum/science/files/CS_in_Science_Module_1.pdf
24	19/11	**Introduction to Simulation and Modeling Languages for Computer Scientists**	**HOMEWORK #5 ASSIGNED** Simpler Spreadsheet Simulation of Multi-Server Queues

		and Engineers Waiting Line (Queue) Models in Excel	
25	24/11	**Introduction to Simulation and Modeling Languages for Computer Scientists and Engineers**	**http://heather.cs.ucdavis.edu/~matloff/156/PLN/DESimIntro.pdf**
26	26/11	**Introduction to Simulation and Modeling Languages for Computer Scientists and Engineers**	**http://heather.cs.ucdavis.edu/~matloff/156/PLN/DESimIntro.pdf**
27	1/12	**Using GPS Location Devices and Android** • Introduction to AppInventor • Introduction to GPS technology	Step by Step AppInventor code.google.com/p/android-scripting https://github.com/damonkohler/sl4a code.google.com/p/android scripting/wiki/SharingScripts
28	3/12	**Using GPS Location Devices and Android** • Introduction to AppInventor • Introduction to GPS technology	Step by Step AppInventor code.google.com/p/android-scripting code.google.com/p/android scripting/wiki/SharingScripts
29	8/12	**Public Holiday – No Class**	
30	10/12	**Using GPS Location Devices and Android** • Introduction to AppInventor • Introduction to GPS technology	**PROJECT #5 ASSIGNED** AppInventor Book code.google.com/p/android scripting/wiki/ApiReference
Final Examination Due			