Fundamentals of Computer Science

Course syllabus

This course introduces computer science through three of its major fields: hardware systems (physical components, digital logic, and computer architecture), theory and algorithms (Boolean algebra, binary arithmetic, and theory of computation), and software systems (languages, compilers, computer graphics, operating systems, and computer networks.) Programming assignments are used as means to introduce and reinforce fundamental computing concepts, as well as computer programming skills that are useful beyond this course. The course provides elements now essential to understand and effectively interact with the information technology infrastructure of today's world.

SESSION	TITLE	Topics	ACTIVITIES	OPTIONAL ACTIVITIES/TOPICS
1	**Course Introduction**	• Course review • General aspects • Class style and policies • Learning material review. • Introduction to CS • Problem solving in CS	➢ Evaluation of preliminary knowledge in Computer Science and Programming ➢ Administrative tasks ➢ Reading: "What is CS" by Eric Suh.	➢ Icebreaker ➢ Group discussion: Learning expectations from the course
Lab	**Interaction with programming environment**	• Python through examples • Creation of simple programs • Instruction(s) for data output	➢ Testing the programming environment. ➢ "Hello world" ➢ Add two numbers. ➢ "The Avatar" (Making figures with characters)	➢ Instruction(s) for data input ➢ Use of variables ➢ Free hands-on activity
2	**Introduction to computer programming**	• Instructions for data output and data input. • Variables and lists • Operators and hierarchy • Boolean expressions • Characters and strings • Conditional and iterative statements • Random number generation • Libraries • User defined functions	➢ Question prompt ➢ "What is your name?" ➢ "What is your age?" ➢ "The Writer" (madlib) ➢ Geometric figures generated by loops (diagonals, back diagonals, triangles, rectangles, perimeters) ➢ "The Fortune Teller"	➢ "99 bottles of root beer" ➢ "12 days of Christmas" ➢ "Rock, paper, scissors." ➢ Common algorithms: Fibonacci, factorial, largest number, average ➢ Flowcharts
Lab	**Binary numbers**	➢ Introduction to ASCII code ➢ Introduction to binary numbers ➢ Programming converters between binary and decimal numbers.	➢ Conversions between binary and decimal numbers ➢ Encoded messages: Exercise on ASCII and	➢ Hexadecimal ←→ Binary

SESSION	TITLE	Topics	ACTIVITIES	OPTIONAL ACTIVITIES/TOPICS
			binary numbers ➢ Bitmapped characters (decimal to binary exercises) **Programming exercises:** ➢ Binary ←→decimal converter. ➢ Bitmapped characters decoders.	
3	**Data representation**	• Representation of numbers • Representation of symbols and strings • Representation of images • Representation of audio	➢ Demonstration of image quality according to amount of bits per pixel. ➢ Demonstration of analog signals vs. digital ones. ➢ Demonstration of audio quality according to amount of bits per sample.	➢ Demonstration of analog audio: vinyl recording. ➢ Demonstration of digital recording: audacity samples. ➢ Binary representation of decimal numbers with fractional part. (Fixed point vs. floating point.)
Lab	**Data manipulation**	➢ Experiments to modify media files to observe impact on quality.	➢ Bitmapped-image viewer ➢ Image editor to modify bits per pixel. ➢ Audio editor to modify sampling rate and bits per sample.	➢ Reading: *The Greatest Mystery in Modern Science,* by Chazelle
4	**Fundamental ideas** • **Boolean algebra**	• Fundamental ideas in CS • Boolean algebra • Truth tables • Binary system: place value system, operations, conversions • Digital circuits • Binary circuits to perform arithmetic operations	➢ Examples of fundamental ideas. ➢ Boolean algebra symbols and examples. ➢ Relationship between Boolean algebra and binary system. ➢ Logic gates ➢ Binary arithmetical and circuits for addition	➢ Reading: *The Rope and Pulley Wonder*, by Dewney.

SESSION	TITLE	Topics	ACTIVITIES	OPTIONAL ACTIVITIES/TOPICS
Lab	**Digital circuits simulation I**	• Introduction to MMLogic (digital circuit simulator)	**Simulation exercises:** ➢ MMLogic experiments ➢ Testing logic gates ➢ Verifying Boolean laws ➢ Implementing binary adders	
5	**Boolean algebra and logic circuits II**	• Combinational logic • Circuits and truth tables • Circuits for arithmetic.	➢ Logic gates and multiplexors ➢ Binary arithmetic: half adder, full adder, 3-bit adder. ➢ Fundamentals of ALU design	➢ Subtraction and two's complement ➢ Video on integrated circuit making.
Lab	**Digital circuits simulation II**	• Binary circuits to perform arithmetic operations	➢ MMLogic experiments	
General Review (Sunday session)	**Movie analysis**	• Computational elements in Disney's "Tron"	➢ CGI, operating systems, CPU, bugs, virus, antivirus, software, simulation, etc.	
6	**Computer architecture**	• Computer parts review • Computer architecture fundamentals • von Neumann model: Stored program concept, data-driven machine, etc.	➢ CARDIAC (educational computer model) ➢ Assembly language exercises.	
Lab	**Computer architecture**	• Implementing CARDIAC virtual machine • Making assembly programs for CARDIAC virtual machine	➢ Absolute value ➢ Integer division ➢ Sign function ➢ Add first *N* naturals	
7	**Numeric-intensive computing I**	• Common series • Numerical approximation by Taylor series	➢ Exercises on numerical series ➢ Approximation of irrational numbers	

SESSION	TITLE	Topics	ACTIVITIES	OPTIONAL ACTIVITIES/TOPICS
Lab		• Implementation of numerical approximation through iterative structures	➢ Harmonic series ➢ Zeno's dichotomy ➢ Approximation of PI, *e*, etc	
8	**Advanced topics I**	• Library research and readings	➢ *WWW: WATCH* by Sawyer (Chapter 1) ➢ *The Rope-And-Pulley Wonder* by Dewney. ➢ *The Turing Omnibus by* Dewney ➢ *Computer Organization* by Feynman ➢ *Why programs fail* by Zeller ➢ *Computer Networks* by Kurose ➢ *Computer Architecture* by Hennesy ➢ *Intelligent machines* by Alan Turing ➢ *The Pattern on the Stone* by Hillis ➢ *Great Ideas in Computer Science* by Biermann ➢ *A.I. for Games* by Fenge	➢ *How to Program by Computer* by Dromey ➢ *The Tinkertoy Computer* by Dewney. ➢ *The Best of Byte* ➢ *Program Design with Pseudocode* by Bailey ➢ *A Hitchhiker Guide to Virtual Reality* by McMenemy ➢ *The Advent of the Algorithm* by Berlinsky ➢ *Creative Evolutionary Systems* by Bentley ➢ *Nine Algorithms that Changed the Future* by MacCormick ➢ *Fundamentals of Natural Computing* by Nunes ➢ *The Limits of Computing* by Walker ➢ *Why Programs Fail* by Zeller
Lab	**Computer generated imagery** (CGI)	• Computer graphics • Memory and CGI • CGI evolution	➢ Movie analysis: Computer generated Imagery (CGI) in Pixar's short films	➢ Raster vs. vector graphics. ➢ Mathematics of 2D images ➢ 3D transformations ➢ Raytracing
9	**Numeric-intensive computing II (Graphics)**	• Graphic mode, • Coordinate systems, • Plotting formulas	➢ Plotting linear and quadratic equations ➢ Parametric curves ➢ Image display	
10	**Advanced topics II**	• Turtle geometry • Recursion • Recursive definitions • Fractal geometry • Introduction to formal grammars	➢ Exercises with the Logo programming language. ➢ Picture generation through formal grammars	➢ Classic geometric exercises solved with turtle geometry ➢ Implementing recursive functions in Logo ➢ Generating fractal shapes through recursion (Snowkfake, Sierpinski, etc.) ➢ Generating grammar-based skylines ("Manhattan skylines")
General Review (Sunday session)	**Advanced topics III**	• Elements of formal grammars • Elements of graph theory	➢ From formal grammars to computer languages ➢ Graph theory apply to maze solving	

SESSION	TITLE	Topics	ACTIVITIES	OPTIONAL ACTIVITIES/TOPICS
		• Cryptography	➢ Caesar's cipher	
11	**Theoretical Computer Science**	• Automata • Cellular automata • Pseudo-random number generation	➢ Abstract notion of graphs, automata and finite state machine ➢ Conway's Game of Life	
Lab	**Final project** (introduction and prototyping)	• Discussion of final projects • Brainstorming • Initial planning • Integration of programming elements required • Research topic	➢ General requirements ➢ Decision-based game: adventure ➢ Virtual pet ➢ Computer electronics and microcontrollers ➢ Mobile computing applications ➢ Games involving simulation or random numbers.	➢ Arduino controlled mobile-robot(C-Language) ➢ Game on MIT App Inventor for Android (visual language)
12	**Theory of computation**	• Computability • Turing machine • Formal definition of algorithm • Algorithm complexity • Operating systems and networking	➢ Simple computers equivalent to Turing machines ➢ Machine language and compilers ➢ Halting problem. ➢ Examples on complexity.	➢ Examples on complexity.
Lab	**Final project development** **(**Hands-on sessions)	• Implementing project major features • Documenting project	➢ Feedback	
13	**Final project development** (Hands-on sessions)	• Implementing project details • Create project context	➢ Preparation of: presentation, webpage, report, etc.	➢ Project review
14	**Final project conclusion**	• Oral report		
15	**Course conclusion**	• Course review.	➢ Final assessment of student's progress	➢

Notes: Each entry provides a general description on the main topic to be studied and activities to be performed. **Topics** indicate specific points to be discussed during the session. In general, **session** refers to in-classroom meetings during the morning. **Labs** are activities to be held in a computer laboratory, and are aimed at promoting student-paced learning and opportunities for individual feedback. Evening session will give students additional time to complete lab work, and also promote analysis of topics presented during the day. Development of personal endeavors will be encouraged during this time.