

**COMPUTER SCIENCE 101
INTRODUCTION TO COMPUTER CONCEPTS**

BULLETIN INFORMATION

CSCE 101: Introduction to Computer Concepts (3 credit hours)

Course Description:

History, application, and social impact of computers; problem-solving, algorithm development, applications software, and programming in a procedural language. Open to all majors.

Prerequisites: Two years of college preparatory mathematics or equivalent

SAMPLE COURSE OVERVIEW

TBA

ITEMIZED LEARNING OUTCOMES

Upon successful completion of Computer Science 101, students will be able to:

1. Demonstrate the ability to find a solution and write an algorithm when given an English description of a task to be accomplished (that is, a problem statement)
2. Demonstrate the ability to write, execute, test, and debug computer programs in a high-level language
3. Demonstrate the mastery and use of the concepts and proper terminology related to computer science

SAMPLE REQUIRED TEXTS/SUGGESTED READINGS/MATERIALS

1. Nell Dale and John Lewis, Computer Science Illuminated, (5th Edition), ISBN 97814496728435.

SAMPLE ASSIGNMENTS AND/OR EXAMS

1. Two Exams and a Final exam
2. Lab
3. Quizzes/Homework

SAMPLE COURSE OUTLINE WITH TIMELINE OF TOPICS, READINGS/ASSIGNMENTS, EXAMS/PROJECTS

Week 1: Introduction to computing, history of computing, the web and the internet, browsers, search engines, URLs, domain registration, E-mail, privacy/security/spam, wikis, social networks, ubiquitous computing, Moore's Law, categories of computers, hardware/software, applications/Excel, computer Languages, generations of programming languages, markup languages, ethical issues - included in each chapter; issues in the news (to be included throughout the semester)

- Week 2:** Creating a web page with HTML5, syntax, elements and tags, nesting, from requirements to design, HTML5 compliant, writing, debugging, testing, using images, adding links, applying style with CSS, format, properties, testing considerations
- Week 3:** Computer Languages, low level languages, assembly language, assembly language code, introduction to binary, high level languages, compilers, assemblers, interpreters, object-oriented overview, problem solving, algorithms, pseudo code , programming logic, programming constructs, program design/testing/debugging, introduction to Alice, objects, properties, methods, events, looping, testing.
- Week 4:** Storage and processing hardware, computer components (disassemble a computer in class or show the pieces from the one already disassembled.) stored program concept/Von Neumann architecture, RAM/ROM, secondary storage devices, touch screens, Non-Von Neumann architecture, solid-state storage devices, perpendicular storage, multi-layer disks, L1/L2 cache, CMOS, performance enhancement techniques -hyper threading, pipelining, bursting, parallel processing, multi-core CPUs, programming logic flow
- Week 5:** Test review, test 1
- Week 6:** Return test, problem solving, data/information, from the problem to the algorithm, pseudo code, moving from the algorithm to writing the code, logic flow, debugging techniques, testing, quality of test data, introduction to CH (C/C++ – Ch interpreter), understanding the language, syntax, variables, data types, storing data for later use, arrays, assignment, retrieving data, I/O, execution, interpreted/compiled, test data, analyzing the output, errors syntax/logical, debugging
- Week 7:** Problem solving, programming examples, programming logic, mathematical calculations. Number systems, back to binary, octal, hexadecimal, conversion, binary arithmetic, data representation, analog, digital, binary representation, ASCII, Unicode, compression, representing numbers/text/audio/graphics/video
- Week 8:** Problems requiring decision making, programming logic flow, conditional statements in Ch, nesting if/else. Networks, cables, throughput, bandwidth, types of networks, internet connections, packet switching, protocols, TCP/IP, MIME types, and addresses , wireless technologies, Wi-Fi, WEP, WPA, WPA2, range, wardriving, Bluetooth, cell phone tech, pings, tracking, unlocking phones, 3G, 4G

- Week 9:** Solving problems requiring more complex decision making, programming using complex conditions in Ch, and/or, logic flow, debugging. System software, operating systems, memory management techniques, process management, CPU scheduling, file systems, directories, utilities, disk cleanup, backup, encryption, compression, defrag, registry cleaners, file shredders, file recovery, download accelerators, etc., software licenses and copyright laws, commercial software, freeware, shareware, open source, abandonware, etc., terms of use, pros and cons of open source, software copyright issues
- Week 10:** Test review, test 2
- Week 11:** Solving problems requiring loops, looping in Ch, types, logic flow, debugging. Data and information, database Systems, database design, DBMS, SQL, features and benefits, database recovery techniques, data mining and its applications
- Week 12:** Problems that require arrays, arrays in Ch, data types, storing and retrieving data, logic flow, Information Security, software design and testing issues, bugs and procedural errors, patches, computer criminals (various types of hackers, professional crime rings, spies, cybercrime, network-based, non-tech attacks, social engineering, insiders, privacy policies and user rights
- Week 13:** Information Security continued, confidentiality, integrity, availability, cryptography, biometrics and identification, physical features, pros and cons, use with RFID tags, password selection rules, security software (antivirus, antispyware, firewalls, intrusion detection/prevention software, etc.), auction fraud, scams, cyber stalking, etc., online resources, legal deterrents, privacy laws, electronic surveillance laws, DRM
- Week 14:** Artificial intelligence, Turing test, knowledge representation, expert systems, Neural Networks, natural language processing, robotics, examples of AI systems. Review for the final exam.