# CSCE 101 - Introduction to Computer Concepts

1. Course number and name: CSCE 101 - Introduction to Computer Concepts

2. Credit: 3-hrs; Contact: 2 lectures of 50 minutes each and a lab per week

3. Instructor: Spring 2012: Catherine T. Matthews, Coordinator

4. Text book: Nell Dale and John Lewis, *Computer Science Illuminated*, (5th *Edition*), ISBN 9781449672843.

5. Specific course information
   a. Bulletin description: History, application, and social impact of computers; problem-solving, algorithm development, applications software, and programming in a procedural language. Open to all majors.
   b. Prerequisites: Two years of college preparatory mathematics or equivalent

6. Learning Outcomes. Specific outcomes of instruction are that 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

7. Topics covered and approximate weight (14 weeks, 3 hours/week, 42 hours total)

   a. History of computers (1 hour)
   b. Programming languages/ Introduction to programming (2 hours)
   c. Problem solving/Algorithms/Pseudo-code (3 hours)
   d. Variables/data types (2 hours)
   e. Logical flow (4 hours)
   f. Debugging/Testing (2 hours)
   g. Conditional logic (if/else, not, and/or) (3 hours)
   h. Loops (2 hours)
   i. Arrays (2 hours)
   j. Number systems/binary/octal/hexadecimal/decimal/conversion/binary arithmetic/data representation (3 hours)
   k. Computers in society/ethics/privacy/benefits/problems (2 hours)
   l. Security/Networks/Database/Artificial intelligence (5 hours)
   m. The web and the internet/History/Browsers/Search engines/Programming for the web (3 hours)
   n. Hardware/Software/Applications (2 hours)
   o. Reviews and Examinations (6 hours)

8. Grade consists of:
   Two Tests 40%

Final Exam 20%
Lab 30%
Quizzes/Homework 10%

Lab Grades

The grade is calculated using the standard curve:
A: 90-100,
B+: 87-89,
B: 80-86,
C+: 77-79,
C: 70-76,
D+: 67-69,
D: 60-66, F:
<60

You must pass (average on must be >= 60) the lab portion of the course in order to pass the class.

9. Specific topics covered.

**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, ethical issues - included in each chapter; issues in the news (to be included throughout the semester), introduction to and the use of HTML5.

**Week 2:** Creating a web page with HTML5 and CSS, 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, categories of computers, hardware/software, applications, computer Languages, generations of programming languages, markup languages.

**Week 3:** Introduction to App Invertor, objects, properties, methods, events, 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,

**Week 4:** Manipulation of variables and preforming calculations using App Inventor, 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:** Introduction to accessing web resources using App inventor, 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, variables, data types, storing data for later use, assignment, retrieving data, I/O, execution, interpreted/compiled, test data, analyzing the output, errors syntax/logical, debugging

**Week 6:** Test review, test 1

**Week 7**: Introduction to sprites and games in App Inventor, 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:** Using the clock to manage events in App Inventor, problems requiring decision making, managing events, 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, Bluetooth, cell phone tech, pings, tracking, 3G, 4G

**Week 9:** Solving more complex problems requiring decision making using App Inventor, 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:** Develop algorithm for final capstone App Inventor project,  Solving problems requiring loops, 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:**  Algorithm to implementation,  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:** Implementation and testing**,** 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:** Converting binary/decimal with App Inventor, Artificial intelligence, Turing test, knowledge representation, expert systems, Neural Networks, natural language processing, robotics**,** examples of AI systems. Review for the final exam.