

Information and Computer Sciences (ICS)

Information and Computer Sciences (ICS) Classes

ICS 101 : Digital Tools for the Information World

Credits: 3

Class Hours: 3 lecture

Recommended: Qualified for ENG 100 and MATH 103. Keyboarding experience is recommended prior to taking this course.

Description: This course covers fundamental information technology concepts and computing terminology, productivity software for problem solving, computer technology trends, and impact on individuals and society. Emphasis will be placed on the utilization of operating systems and the production of professional documents, spreadsheets, presentations, databases, and web pages.

Semester Offered: Fall, Spring

Course Student Learning Outcomes (CSLOs):

1. Utilize operating system interfaces to manage computer resources effectively.
2. Use accepted hardware, software, and communications terminology to effectively interact with other computer users.
3. Know the ethical and security issues involved in the use of computer technology.
4. Utilize appropriate computer applications to produce professional-level documents, spreadsheets, presentations, databases, and web pages for effective communication.
5. Extract and synthesize information from available Internet resources using intelligent search and discrimination.

ICS 111 : Introduction to Computer Science I

Credits: 3

Class Hours: 3 lecture

Prerequisites: Qualified for MATH 103.

Recommended: Basic computer use proficiency.

Description: This course is intended for Computer Science majors and all others interested in the first course in programming. This course is an overview of the fundamentals of computer science emphasizing problem solving, algorithm development, implementation, and debugging/testing using an object-oriented programming language.

Semester Offered: Fall, Spring

Course Student Learning Outcomes (CSLOs):

1. Illustrate basic programming concepts such as program flow and syntax of a high-level general purpose language
2. Identify relationships between computer systems, programming and programming languages.
3. Demonstrate working with primitive data types, strings, and arrays.
4. Use an appropriate programming environment to design, code, compile, run, and debug computer programs.
5. Demonstrate basic problem solving skills using algorithms in an object-oriented computer language.