ICS 140 : Computational Thinking with Programming

A. Course Description

Credits: 4

Prerequisites:
- MATH 115 College Algebra
- OR
- MATH 120 Precalculus
- OR
- STAT 201 Statistics I

Lab Hours/ Weeks: Corequisites: None

Lecture Hours/ Week:

MnTC Goals: None

An introduction to the formulation of problems and developing and implementing solutions for them using a computer. Students analyze user requirements, design algorithms to solve them and translate these designs to computer programs. The course also provides an overview of major areas within the computing field. Topics include algorithm design, performance metrics, programming languages and paradigms, programming structures, number representation, Boolean algebra, computer system organization, data communications and networks, operating systems, compilers and interpreters, cloud computing, data analytics, mobile computing, internet of things, and artificial intelligence) database, internet, security, privacy, ethics, and other societal and legal issues. Lab work and homework assignments involving flow charting tools and programming using a language such as Python form an integral part of the course.

B. Course Effective Dates: 08/15/2017 - Present

C. Outline of Major Content Areas:

See Course Description for major content areas.

D. Learning Outcomes (General)

1. Describe algorithms and their role in solving problems.
2. Explain the role of the operating system in a digital system.
3. Describe emerging technologies such as cloud computing, artificial intelligence, data analytics and others.
4. Describe different types of application software including database, internet, and games.
5. Explain security and privacy threats and describe measures to prevent them.
6. Describe the evolution of digital systems and their past, present, and potential future impact on human society.
7. Develop algorithmic solutions using appropriate programming structures and express algorithms using pseudocode, flow-charts or other design notation.
8. Demonstrate knowledge of common algorithms, such as binary search, finding the minimum/maximum value in a list, and quadratic sort.
9. Demonstrate the ability to select an appropriate algorithm for solving a problem.
10. Implement algorithms with a high-level programming language and provide simple documentation.
11. Express numbers in binary format.
12. Express and simplify Boolean expressions.
13. Describe the Von Neumann architecture and interaction between the processor and memory.
14. Explain how data communication works and give examples of networks.

E. Learning Outcomes (MN Transfer Curriculum)
G. Special Information

Note: Students must be proficient in using a computer. Required skills include the ability to create directories (folders) and files of different types (plain text files, zip files, etc.) using a graphical user interface as well as a command line interface. Students are responsible to both be aware of and abide by prerequisites for ICS courses for which they enroll, and will be administratively dropped from a course if they have not met prerequisites.