Computational Core – an Online Undergraduate Certificate in Computer Science

Effectively using computers to solve problems is a key part of the modern, technology-driven workplace. Build your portfolio of career skills by completing the Computer Science online undergraduate certificate.

The certificate includes 5 programming courses and 17 credit hours that teach you the skills you need to solve real-world problems by developing your own computer software. Each course is delivered in a state-of-the-art online computer science education environment that lets you practice what you are learning as you learn it.

Core Requirements (17 Hours)

CC 110 - Introduction to Computing (3 credits, fall/spring/summer)

An overview of the history of computers and programming. Famous historical figures and their impact in modern computing. Introduction to major topics in computer science such as artificial intelligence, high performance computing, cryptography, big data, cybersecurity, robotics, and more. Brief experience with computer programming concepts.

CC 210 - Fundamental Computer Programming Concepts (4 credits, fall/spring/summer)

Basic concepts in developing computer programs: program structure and syntax, primitive data types, variables, control flow, iteration, simple algorithms, debugging, and good software development practices. Introduction to object-oriented programming.

The course introduces students to computer programming using one of several programming languages. Interactive lessons and engaging projects reinforce new skills and concepts while relating programming fundamentals to the real world. This course covers the basic concepts of programming, from variables and control flow to functions, objects, and simple algorithms.

CC 310 - Data Structures & Algorithms 1 (3 credits, fall/spring/summer)

Exploration of simple data structures & related algorithms in computer programming. Basic concepts of complexity analysis. Object-oriented design concepts.

This course introduces simple data structures such as sets, lists, stacks, queues, and maps. Students learn how to create data structures and the algorithms that use them. Students are introduced to algorithm analysis to determine the efficiency of algorithms.

CC 315 - Data Structures & Algorithms 2 (3 credits, fall/spring/summer)

The course covers more advanced data structures & related algorithms. More focus is placed on formal software development methods and software engineering fundamentals. The course also includes an introduction to requirements analysis processes that provides the specification of algorithmic requirements.

This course introduces advanced data structures, such as trees, graphs, and heaps. Several new algorithms using these data structures are covered. Students also learn software development methods and software engineering fundamentals and use those skills to develop projects of increasing size and scope effectively.

CC 410 - Advanced Programming (4 credits, fall/spring/summer)

Advanced programming techniques and projects. Concepts from simulation and modeling, media applications, secure design, information management, parallelism, and networking. Software development methodologies, processes, and design patterns. Practical experience with professional communication and collaboration.

In this course students gain experience writing programs using a variety of advanced programming techniques. Projects cover a variety of application domains and use a variety of technologies to help students master advanced computer programming concepts.

For More Information, visit: http://www.cs.ksu.edu/core/

Department of Computer Science
Carl R. Ice College of Engineering
Kansas State University
2184 Engineering Hall
1701D Platt St.
Manhattan, KS 66506
785-532-6350
csoffice@k-state.edu
cs.ksu.edu