

C S-COMPUTER SCIENCE (C S)

C S 111. Computer Science Principles

4 Credits (3+2P)

This course provides a broad and exciting introduction to the field of computer science and the impact that computation has today on every aspect of life. It focuses on exploring computing as a creative activity and investigates the key foundations of computing: abstraction, data, algorithms, and programming. It looks into how connectivity and the Internet have revolutionized computing and demonstrates the global impact that computing has achieved, and it reveals how a new student in computer science might become part of the computing future. May be repeated up to 4 credits.

Prerequisite: MATH 1215 or higher.

Learning Outcomes

1. Identify and differentiate programming constructs like IF, FOR, and WHILE
2. Convert numbers between Hexadecimal, Binary and Decimal
3. Write pseudo code to manipulate a robot
4. Use an ASCII table to translate HEX strings into characters
5. Encrypt and Decrypt simple messages with a Caesar Cypher

C S 117. Introduction to Computer Animation

3 Credits (3)

Introductory course for learning to program with computer animation as well as learning basic concepts in computer science. Students create interactive animation projects such as computer games and learn to use software packages for creating animations in small virtual worlds using 3D models. Recommended for students considering a minor/major in computer science or simply interested in beginning computer animation or programming.

C S 151. C++ Programming

3 Credits (2+2P)

Introduction to object-oriented programming in the C++ language. The focus will be on preparing students to use C++ in their own areas. No prior programming experience is required. Taught with C S 451.

Prerequisite: MATH 1215 or higher.

Learning Outcomes

1. Use various data types and the corresponding operations.
2. Write C++ programs that contain expressions, program control, functions, arrays, and input/output
3. Explain basic object-oriented programming concepts.
4. Demonstrate proficiency in using classes, inheritance, pointers, streams, and recursion

C S 152. Java Programming

3 Credits (2+2P)

Programming in the Java language. May be repeated up to 3 credits.

Prerequisite(s): MATH 1215 or higher.

C S 153. Python Programming I

3 Credits (3)

This course is an introduction to programming in the Python language, covering fundamental scripts, data types and variables, functions, and simple object creation and usage. The focus will be on preparing students to use Python in their own areas. No prior programming experience is required. Taught with C S 453.

Prerequisite: MATH 1215 or higher.

Learning Outcomes

1. Develop an algorithm to solve a problem
2. Demonstrate the ability to use Python data types: int, float, strings, and lists; and the built-in functions associated with those data types
3. Edit and debug programs using the Spyder IDE for Python
4. Implement algorithms using the Python features of assignment, input, output, branches, loops, and functions
5. Explain the fundamental concepts of object-oriented programming with Python
6. Design and implement Python classes based on given attributes and behaviors
7. Work with existing Python modules such as math, random, and os
8. Write Python programs that input data from files and store results in files

C S 154. Python Programming II

3 Credits (3)

This course covers advanced Python programming, including classes, objects, and inheritance, embedded programming in domain applications, database interaction, and advanced data and text processing. The focus will be on preparing students to use Python in their own areas.

Prerequisite(s): C S 153 or C S 453.

C S 158. R Programming I

3 Credits (3)

This course is an introduction to data processing in the R language, covering fundamental script configuration, data types and data collections, R control structures, and basic creation of graphs and data visualizations. This course will not focus on the statistical capabilities of R, though some basic statistical computations will be used.

Prerequisite(s): MATH 1220G.

C S 171G. Modern Computing in Practice

4 Credits (3+2P)

This course provides a survey of practical and theoretical foundations for how computers work and how they support fundamental organizational needs. The course covers broad aspects of the hardware, software, and mathematical basis of computers. Lab assignments provide hands-on applications to use simple basic software tools to write simple programs, build and edit websites, analyze data with spreadsheets, choose an office productivity suite, and demonstrate computer literacy to potential employers. May be repeated up to 4 credits.

Prerequisite: MATH 1130G or MATH 1215 or higher.

Learning Outcomes

1. Students will create simple python programs using conditional statements and loops
2. Students will analyze data with spreadsheet formulas, charts, and tools
3. Students will create and publish a personal website using website building tools
4. Students will edit HTML and CSS to format a website manually
5. Students will practice the skill of performing software QA and providing actionable feedback to developers
6. Students will become aware of common cybersecurity risks
7. Students will learn basic vocabulary and context for broad aspects of hardware, software, and computer science theory such as Security, Privacy, Cloud Computing, the Internet, the Web, Operating Systems, Discrete Math, and Information Systems
8. Students will be exposed to various sub-fields of CS including artificial intelligence, security, data analytics, UX, web development, and QA testing

9. Students will reason about the societal impacts of technology 1
10. Students will incorporate their new knowledge and skills into their resume

C S 172. Computer Science I**4 Credits (3+2P)**

Computational problem solving; problem analysis; implementation of algorithms using Java. Object-oriented concepts, arrays, searching, sorting, and recursion. Taught with C S 460. May be repeated up to 4 credits.

Prerequisite: (A C- or better in either MATH 1250G or (MATH 1430G or higher)) OR (A C- or better in MATH 1220G and a 1 or better in the CS Placement Test) OR (A C- or better in MATH 1220G and a C- or better in C S 111).

Learning Outcomes

1. Develop algorithms to solve problems
2. Implement algorithms using the fundamental programming features of sequence, selection, iteration, and recursion
3. Apply an understanding of primitive and object data types
4. Design and implement classes based on given attributes and behaviors
5. Explain the fundamental concepts of object-oriented programming,

C S 209. Special Topics.**1-3 Credits**

May be repeated for a maximum of 12 credits.

C S 271. Object Oriented Programming**4 Credits (3+2P)**

Introduction to problem analysis and problem solving in the object-oriented paradigm. Practical introduction to implementing solutions in the C++ language. Pointers and dynamic memory allocation. Hands-on experience with useful development tools. Taught with C S 462. May be repeated up to 4 credits.

Prerequisite: At least a C- in C S 172 or ENGR 140.

Learning Outcomes

1. Develop an algorithm to solve a problem.
2. Implement algorithms using the C and C++ languages including imperative and object-oriented language features.
3. Beyond what was learned in C S 172, E E 112, or E E 161 demonstrate a noticeable increase in understanding of problem analysis and program design.
4. Demonstrate proficiency in using control structures including if statements (single selection), switch (multiple selection), and loops (repetition).
5. Demonstrate proficiency in using arrays and functions
6. Create UML class and relationship diagrams.
7. Design a class to model a real-world person, place, thing, or event.
8. Use editing and debugging software to create, debug, and test C and C++ programs.
9. Understand the basic terminology used in object-oriented programming. 1
10. Create a make file to build an executable from a set of C or C++ source files.

C S 272. Introduction to Data Structures**4 Credits (3+2P)**

Design, implementation, use of fundamental abstract data types and their algorithms: lists, stacks, queues, deques, trees; imperative and

declarative programming. Internal sorting; time and space efficiency of algorithms. Taught with C S 463.

Prerequisite: At least a C- in C S 172, or placement.

Learning Outcomes

1. Be able to implement and use lists
2. Be able to implement and use stacks
3. Be able to implement and use queues
4. Be able to implement and use trees
5. Be able to perform the run time analysis of basic algorithms using Big O notation
6. Be able to implement, use, and analyze searching algorithms
7. Be able to solve a problem recursively
8. Take a problem statement from a user and convert it into a Java program that fulfills the user's needs
9. Create object oriented Java classes that effectively separate and hide implementation details from client applications

C S 273. Machine Programming and Organization**4 Credits (3+2P)**

Computer structure, instruction execution, addressing techniques; programming in machine and assembly languages. Taught with C S 464. May be repeated up to 4 credits.

Prerequisite: At least a C- in C S 172 or ENGR 140.

Learning Outcomes

1. Describe the architecture of a microcontroller, the interconnections between the components, and the basic units inside the CPU
2. Use signed and unsigned numbers, the associated branching instructions, and the corresponding flags in the status register
3. Explain immediate, direct, indirect addressing modes, their opcode and operands, and their utilities
4. Map high-level programming language features to assembly instructions, including loops, conditionals, procedure calls, value and reference parameter passing, return values, and recursion
5. Interface with I/O devices including LED and sensors via digital input and output, and analog-to-digital conversion
6. Program timers/counters and interrupts to control real-time applications
7. Design an assembly program

C S 278. Discrete Mathematics for Computer Science**4 Credits (3+2P)**

Discrete mathematics required for Computer Science, including the basics of logic, number theory, methods of proof, sequences, mathematical induction, set theory, counting, and functions. Taught with C S 465.

Prerequisite: At least C- in C S 172.

Learning Outcomes

1. Use logic to specify precise meaning of statements, demonstrate the equivalence of statements, and test the validity of arguments
2. Construct and recognize valid proofs using different techniques including the principle of mathematical induction
3. Use summations, formulas for the sum of arithmetic and geometric sequences
4. Explain and apply the concepts of sets and functions
5. Apply counting principles to determine the number of various combinatorial configurations