B.A. in COMPUTER AND INFORMATION SCIENCE – Computational Mathematics

(32 credits required for graduation with a minimum cumulative GPA of 2.00)

NOTE: This guide is not meant to replace the degree audit; it is subject to change and represents actions approved by Faculty to date. Students are encouraged to run their degree audit at the end of each term of enrollment. Please refer often to the *22019-2020 Online Catalog & Student Handbook (http://catalog.berea.edu/en/current/catalog),* which will be updated with the most current information.

GENERAL EDUCATION PROGRAM

No single transfer course can meet more than one General Education requirement.

Core Courses

(Development math courses may be waived on basis of test scores.)
MAT 010 Pre-Algebra
MAT 011 Elementary Algebra
MAT 012 Elementary Algebra II

GSTR 110 Writing Seminar I: Critical Thinking in the Liberal Arts
(Transfer students may waive if College Composition was taken as a degree-seeking student at another college and earned a grade of B or higher.)
GSTR 210 Writing Seminar II: Identity and Diversity in the U.S.
GSTR 310 Understandings of Christianity
GSTR 410 Seminar-Contemporary Global Issues

Scientific Knowledge and Inquiry

GSTR 332 Scientific Origins **OR**
Two (2) approved science courses, from two different disciplines, one of which must be an approved lab course. The following courses have been approved to meet this requirement: ANR 110, BIO 100, 101, 110, CHM 113, 131, PHY 111, 127, 221

Wellness & Fitness

WELL 101 Principles of Wellness I
WELL 102 Principles of Wellness II
Two (2) ¼-credit HHP activity courses *(HHP 200 will satisfy both the SWIM requirement and one of the activity course requirements)*

Practical Reasoning (PR &PRQ)

Two (2) courses, at least one firmly grounded in math or statistics (PRQ); the other can be an approved practical reasoning (PR) course or another PRQ course.

Perspectives (Six areas required)

One (1) course in **each** of the six areas is required. Individual courses may be approved to satisfy more than one perspective, but no single course may satisfy more than two perspective areas.
1) Arts
2) Social Science
3) Western History
4) Religion
5) African American/Appalachian/Women
6) International (choose one option):
A) Two (2) courses in the same non-English language, one of which may be waived through testing; **OR**
B) Two (2) world culture courses, one of which must be grounded in a non-western culture

Active Learning Experience

An approved experience, taken for credit or non-credit (e.g. internships, undergraduate research experiences).

MAJOR COURSES

Core Courses
CSC 226: Software Design & Implementation
CSC 236: Data Structures

Exploratory Distribution (2 courses)

CSC 110 Craft of Computing
CSC 111 Storytelling-Comp Animation
CSC/BUS 114 Business App & Prog
CSC 121 Introduction to Game Design
CSC 124 Building Better Apps
CSC 126 Intro to Robotics
SENS 320 Intro-Geographical Info System
TAD 130 Design & Documentation
TAD 180 Graphic Com & Design
TAD 330 Computer Aided Drafting & Design
TAD 382 Advanced Graphics
TAD 455 Comp Integrated Manufacturing
TAD 460 Digital Electronics
Any upper-level CSC course not needed to meet another requirement in CIS major

NOTE: Students who have successfully completed CSC 236 *Data Structures* cannot get credit for future offerings of CSC 100-level courses with th eexception of BUS/CSC 114 and future offerings of CSC 186.

Upper-level Distribution (4 total credits)

At least one course chosen from each of the following categories and at least one at the 400-level.

Design
CSC 330 Database Systems
CSC 350 Computer Security
CSC 410 Computational Intelligence
CSC 420 Programming Languages
CSC 426 Open Source Software Engineering

Foundations
CSC 303 Theory of Computation
CSC/MAT 433 Numerical Analysis
CSC 440 Design & Analysis of Algorithm
CSC 445 Computational Complexity & Modeling

Systems
CSC 335 Computer Organization
CSC 412 Networking
CSC 425 Operating Systems & VMs

Capstone (1/2 or 1 credit)

CSC 493 Senior Project
CSC 495 Internship
UGR 010 or 020 Undergraduate Research (as approved by the CIS department)

Collateral Courses (7 total credits)
MAT 135 Calculus I (required)
MAT 225 Calculus II (required)
CSC/MAT 433 Numerical Analysis (required)

Discrete Mathematics (2 courses)
MAT105 Intro to Discrete Math
MAT 214 Linear Algebra
MAT 312 Operations Research
MAT 415 Combinatorics

Upper Level (1 course)
MAT 330 Calculus III
MAT 311 Probability
MAT 315 Fundamental Concepts of Math

Electricity and Electronics (1 course)
TAD 265 Electricity & Electronics
TAD 460 Digital Electronics

ELECTIVES
20 credits outside the major

Learning Goal 1: Students think critically and creatively to solve local or global problems. In particular, successful students should be adept at formulating, analyzing, decomposing, and solving problems computationally. They have internalized the software design and development process, and it has become a "habit of mind" which they can apply in new situations.

Learning Outcome 1.1: Show innovation and creativity in their approach to problem-solving and design.
Learning Outcome 1.2: Identify tasks that are well-suited to be solved computationally.
Learning Outcome 1.3: Apply software design and development process to novel situations.
Learning Outcome 1.4: Analyze and decompose novel problems into components appropriate to the design of an algorithmic solution.
Learning Outcome 1.5: Implement reasonably efficient computational solutions.
Learning Outcome 1.6: Implement, test, and debug computational solutions utilizing different computer platforms.

Learning Goal 2: Students are ready to be successful practitioners in both applied and theoretical areas. Namely, they understand the fundamental concepts and theories of the discipline of computer science as specified by professional organizations such as the Association for Computing Machinery (ACM) and can apply them.

Learning Outcome 2.1: Employ in-depth principles of design to at least one important area of application.
Learning Outcome 2.2: Demonstrate facility with one widely used programming language and at least two other languages, at least one of which is a systems language such as C or C++.
Learning Outcome 2.3: Articulate and apply theoretical and mathematical foundations of computer science, such as algorithm efficiency and computational complexity.
Learning Outcome 2.4: Demonstrate knowledge of the relationship between computer architectures and software systems or operating systems, the software that manages computer hardware resources and provides common services for computer programs.

Learning Goal 3: Appreciate the importance of effective communication and collaboration with both technical and non-technical constituents.

Learning Outcome 3.1: Demonstrate ability to work effectively individually, in pairs, small groups, and larger teams.
Learning Outcome 3.2: Communicate effectively in a range of media with audiences whose background range from technical to non-technical.
Learning Outcome 3.3: Write understandable technical documents that describe the specification, design, and implementation of computational projects.
Learning Outcome 3.4: Effectively elicit technical requirements from clients.

Learning Goal 4: Express potential for lifelong learning and good citizenry.

Learning Outcome 4.1: Develop strategies for continued learning in computer science in a rapidly changing discipline.
Learning Outcome 4.2: Read and assimilate technical material independently from textbooks, articles, and other level-appropriate sources.
Learning Outcome 4.3: Consider the ethical and social impacts of technology to become positive contributors to society.
Learning Outcome 4.4: Be prepared to pursue advanced studies in the field and/or to assume professional responsibilities.