1 Major Requirements

The Computer Science major consists of 13 required courses. No electives are required, though elective material is made available to students through CSCI 70, CSCI 198, and CSCI 199. Each course is a four credit hour offering. A total of 52 hours is required to complete the major. The major requirements are as follows:

- CSCI 1: Introduction to Computers and Computing
- CSCI 2: Object-Oriented Programming
- MATH 3: Introduction to Statistics
- CSCI 10: Human Interaction with Technology and Information
- CSCI 23: Discrete Mathematics for Computer Science
- CSCI 25: Data Structures
- CSCI 100: Systems Programming and Tools
- CSCI 101: Algorithm Analysis and Computability
- CSCI 124: Operating Systems
- CSCI 130: Information Management
- CSCI 140: Software Engineering
- CSCI 150: Net-Centric Computing
- CSCI 160: Capstone - Applications of Computing to Other Disciplines

2 Course Descriptions


CSCI 2: Object-Oriented Programming: Designing, writing, and testing structured computer programs. Decomposing problems; writing function definitions; conditional and iterative control constructs; using class libraries. Problem-solving through programming with classes and vectors; algorithm correctness; recursion. Java will be the language of instruction. **Prerequisite: C- or better in CSCI 1**

CSCI 10: Human Interaction with Technology and Information: A study of how people perceive technology and the ways in which they consume and create information. An introduction to the practice of designing technology with careful consideration for its users. No programming experience is required prior to taking this course.

CSCI 23: Discrete Mathematics for Computer Science: Mathematics central to the study of computer science. Topics include: set theory, logic, induction, combinatorics, number theory, graph theory, sequences and series, matrices, and recurrence relations. **Prerequisite: C- or better in CSCI 1**
CSCI 25: Data Structures: Introduction to the study of abstract data types and the analysis of algorithms. Students will write Java applications using data structures such as linked lists, stacks, queues, multidimensional arrays, trees, sets, maps, and heaps. **Prerequisite:** C- or better in CSCI 2

CSCI 70: Computing Technology, Society, and Culture: This course will survey the principal computing technologies that are in use today or on the horizon, then investigate individual topics in more technical and cultural depth. Topics will vary in light of new developments, and could include blogging, RFID, intelligent systems, GPS, data mining, Google, and eBay. Other aspects of computing technology, society, and culture to be addressed will include legal and political issues such as regulation, jurisdiction, internationalization, and standardization, and broader questions such as how and why new computing technologies are developed and accepted. Enrollment priority: Priority is given to Juniors and Seniors. **Prerequisite:** Sophomore standing required.

CSCI 100: Systems Programming and Tools: Development of software in the C programming language. User-level functionality of the UNIX operating system. Architecture of the UNIX operating system from a programmer’s perspective. Machine-level representation of data; assembly-level machine organization. Tools for large-scale software engineering including integrated development environments and code versioning systems. **Prerequisite:** C- or better in CSCI 2


CSCI 124: Operating Systems: The fundamentals of operating systems design and implementation. Basic structure; synchronization and communication mechanisms; implementation of processes, process management, scheduling, and protection; memory organization and management; file systems; machine-level representation of data; assembly-level machine organization; functional organization of computers. **Prerequisite:** C- or better in CSCI 100.

CSCI 130: Information Management: Theory and practice of information storage, management and retrieval, emphasizing relational database management systems. Case studies of small-scale (personal computing) and large-scale (corporate records on distributed systems) applications. Data modeling, database design and management, query processing, data integrity, and security. Legal and social contexts of data management; the responsibility of professionals to understand requirements, risks, and liabilities. **Prerequisite:** C- or better in CSCI 2 and CSCI 10.

CSCI 140: Software Engineering: Software design; using APIs; software tools and environments; software processes; software requirements and specifications; software validation; software evolution; software project management; methods and tools of working in teams; social context of computing; professional and ethical responsibilities; risks and liabilities of computer-based systems. **Prerequisite:** C- or better in MATH 3, CSCI 10 and CSCI 100.
CSCI 150: Net-centric Computing: Communication and networking; the social context of computing; intellectual property; network security; the web as an example of client-server computing; building web applications; network management; compression and decompression; wireless and mobile computing; virtual machines; knowledge representation and reasoning. **Prerequisite:** CSCI 23 and C- or better in MATH 3, CSCI 10, CSCI 100.

CSCI 160: Capstone - Applications of Computing to Other Disciplines: Much of computer science is practiced through application of computing to other disciplines. In this capstone course, the instructor and students will develop a software solution to a problem arising in another field. Application areas include, but are not limited to finance, economics, biology, and law. We will explore strategies for learning in and contributing to inter-disciplinary teams, customer-client communication; software design, requirements, specification, and project management. **Prerequisite:** CSCI 23, CSCI 25, and C- or better in MATH 3, CSCI 10, and CSCI 100.

CSCI 198: Topics in Computer Science: Topics to be determined by current events in computing and opportunities presented by visiting faculty, etc. Offered fall of even numbered years. **Prerequisite:** Dependent on topic.

CSCI 199: Independent Study in Computer Science: An independent investigation of a topic selected in conference with the instructor and approved by the department. Admission by petition to or invitation from the department. Amount of credit established at time of registration. May be repeated for credit with the approval of the department. Signature of instructor required for registration.

### 3 Computer Science Minor Requirements

The requirements for the minor in Computer Science are as follows. 24 hours are required to complete the minor.

- CSCI 1: Introduction to Computers and Computing
- CSCI 2: Object-Oriented Programming
- CSCI 10: Human Interaction with Technology and Information
- CSCI 23: Discrete Mathematics for Computer Science or CSCI 25: Data Structures
- CSCI 100: Systems Programming and Tools
- One additional upper level course in Computer Science.
4 Prerequisite Structure

Figure 1 summarizes the dependencies between courses in the proposed curriculum.

Figure 1: Course dependencies.
5 Learning Objectives

The following describe the learning objectives the computer science faculty has for graduates of our program.

1. Foundational knowledge: Demonstrate knowledge and understanding of essential facts, concepts, principles, and theories relating to computer science and software applications.

2. Modeling and implementation: Deploy appropriate theory, practices, and tools for the specification, design, implementation, and evaluation of computer-based solutions.

3. Numeracy: Understand, explain, and determine the quantitative dimensions problems and systems.

4. Tools: Deploy effectively, information applications and systems, software development environments, and other tools commonly used in modern computing practice.

5. Evaluation and testing: Evaluate the extent to which a computer-based system meets the criteria defined for its current use and future development. Test systems in terms of general quality attributes, correctness, and possible tradeoffs presented within the given problem.

6. Requirements Analysis: Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.

7. Information management: Apply the principles of effective information management, information organization, and information-retrieval to information of various kinds (e.g., text and structured data).

8. Human-computer interaction: Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, and multimedia systems.

9. Communication: Make succinct presentations to a range of audiences about technical problems and their solutions. Actively listen to customers, team members, and others with whom one works for effective exchange of information.

10. Professional development: Keep abreast of current developments in the discipline to continue one’s own professional development.

11. Social context of computing: Demonstrate knowledge and understanding of the opportunities, implications, and risks of computing in a connected society and technologies that bear on each of these issues.

12. Professional responsibility: Recognize and be guided by the social, professional, and ethical issues involved in the use of computer technology. Manage one’s own learning and development, including time management and organizational skills.

13. Project management: Be able to work effectively as a member of a development team. Knowledge and understanding of the entire software development lifecycle.
<table>
<thead>
<tr>
<th>Learning Objective</th>
<th>MATH</th>
<th>1</th>
<th>2</th>
<th>10</th>
<th>25</th>
<th>28</th>
<th>100</th>
<th>124</th>
<th>130</th>
<th>140</th>
<th>150</th>
<th>160</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foundational knowledge</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
<tr>
<td>Modeling and implementation</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>**</td>
<td>***</td>
</tr>
<tr>
<td>Numeracy</td>
<td>***</td>
<td>*</td>
<td>*</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
<tr>
<td>Tools</td>
<td>**</td>
<td>*</td>
<td>*</td>
<td>**</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
<tr>
<td>Evaluation and testing</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Requirements analysis</td>
<td>**</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Information management</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Human-computer interaction</td>
<td>***</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Communication</td>
<td>**</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Professional development</td>
<td>**</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Social context of computing</td>
<td>**</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Professional responsibility</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Project management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: A summary of the degree to which each course in the curriculum addresses each of the learning objectives. Each cell indicates the intensity with which a learning objective is addressed in a particular course. The scale used is zero to three stars, with three indicating a high level of coverage and zero indicating only incidental coverage.