Computer Science

Andrew J. Crump, Instructor
Brett W. Huffman, Assistant Professor
Clinton A. Staley, Professor
James A. Waller, Assistant Professor

The science of computing is a rapidly growing field of study. Computers are used in almost every facet of life, and a grasp of their fundamentals can lead to challenging and rewarding activity in many fields. Principia’s computer science courses stress the principles that underlie computing as well as practical experience in creating software and problem solving.

Students develop programming skills from beginning to advanced levels and also learn about the fundamental design and architectural considerations of computer systems. In the process, students have plenty of opportunities to demonstrate the discipline and persistence needed to bring a program to completion—all the while gaining confidence in their ability to think systematically and to recognize everyone's natural ability to do so.

Because of the foundational nature of CSCI 182 Intro to Data Structures, a student must earn a grade of C or better in this course (or get permission from the department) to continue as a computer science major. To complete a major or minor offered by the Computer Science Department, a student must earn a cumulative average of 2.000 or higher for the courses (required and elective, within the department or outside it, such as math and engineering) taken for that program.

For all majors and minors, if CSCI 181 Accelerated Intro Programming is substituted for CSCI 171 Introduction to Programming, the total semester hours required will be two fewer.

In addition to both a B.S. and a B.A. in computer science, the Computer Science Department offers two minors, one in computer science and a second in management information systems. The minor in management information systems is designed to prepare students to work with business programming, emphasizing a professional approach. Students in this minor will study methodologies for complex software system development, database management, database design, and professional ethics. Further, an Individualized major and minor is available in Computer Engineering.

A student may earn only one major or minor in computer science.

Departmental Learning Outcomes

1. Design: Our students will develop computer programs and systems characterized by good design: elegance, simplicity, clear structure, flexibility, efficiency, usability, maintainability, ease of use, and durability.

2. Problem Solving: Our students will accurately assess problems and think about them creatively, conceptually, critically, insightfully, analytically, and metaphysically. This will lead them to create, apply, and integrate appropriate and efficient algorithms and mathematical models.
3. **Application of Technical Knowledge:** Our students will understand a sufficiently comprehensive, foundational body of knowledge such that missing pieces can be derived, acquired, or revealed, and applied properly. This body of knowledge consists primarily of computer science topics such as programming, computer hardware and software organization, operating systems, database management, and networking, but also includes elements of information systems and software engineering.

4. **Effective Communication:** Our students will convey and receive information and ideas accurately, consistently, and efficiently in oral, visual, and written form, formally and informally across a diversity of audiences and organizational levels. Good communication necessitates honesty and effective listening.

5. **Teamwork and Collaboration:** Our students will work within diverse groups of people toward common goals by expressing effective communication, resourcefulness, confidence, and humility, while respecting and valuing the contributions of others. Our students will comprehend the qualities of both a true leader and a true follower, and develop the moral courage to lead with authority, thereby bringing out the best in others.

6. **Ability to Learn:** Our students will develop the desire and ability to continuously learn and improve their knowledge, attitudes, and methods in both individual and group settings. They will display accurate self-assessment, wise self-correction, and a growing mastery of their field.

7. **Ethics:** Our students will recognize ethical issues, identify needed actions, and demonstrate the moral courage to implement them. Our students will understand and follow generally accepted codes of conduct such as the [ACM Code of Ethics and Professional Conduct](http://www.acm.org/about/codeofethics) and the [ACM/IEEE Software Engineering Code of Ethics and Professional Practice](http://www.acm.org/about/codeofethics).

8. **Professionalism:** Students will hold themselves responsible for depth of knowledge and excellence of action. Those who best demonstrate professionalism do so by keeping a high quality of thought, meeting obligations, being accountable, giving attention to detail, being dependable and consistent, maintaining a balanced perspective, and respecting others.

**Majors**
- B.S. in Computer Science ([http://catalog.principiacollege.edu/majors-minors/computer-science/bs](http://catalog.principiacollege.edu/majors-minors/computer-science/bs))
- B.A. in Computer Science ([http://catalog.principiacollege.edu/majors-minors/computer-science/ba](http://catalog.principiacollege.edu/majors-minors/computer-science/ba))

**Minors**
- Minor in Computer Science ([http://catalog.principiacollege.edu/majors-minors/computer-science/Minor](http://catalog.principiacollege.edu/majors-minors/computer-science/Minor))
- Minor in Management Information Systems ([http://catalog.principiacollege.edu/majors-minors/computer-science/Minor-management-information-systems](http://catalog.principiacollege.edu/majors-minors/computer-science/Minor-management-information-systems))

**CSCI 170 Intro to Computer Science**

Modern computing concepts, components, history from the abacus to the modern PC. Introduction to computer arithmetic, logic, circuits, a low-level language and a high-level language, elementary systems, peripherals, networks and ethics. Encourages presentation development and delivery. Addresses the fear of technology. Does not teach computer applications (such as word processing) but explores the science of computing. Does not count toward the CSCI major or the MIS minor.
CSCI 171  Introduction to Programming  4.0 SH  [GESN]
Focus on fundamental concepts of problem solving by analyzing problems and using a computer
language (C++, Java, or Python) as a tool to design, code, document, and test solutions, learned
through textbook study, lectures, examples, and considerable practice in the computer science
programming laboratory. Professional approaches are emphasized. (A student may receive credit for
only one of the courses CSCI 171 or CSCI 181.).

CSCI 181  Accelerated Intro Programming  2.0 SH  [ ]
Students entering the computer science program with considerable programming experience may take
this course as a replacement for CSCI 171 as a major requirement.
**Prerequisite:** successful completion of a programming competency test. (A student may receive credit
for only one of the courses CSCI 171 or CSCI 181.).

CSCI 182  Intro to Data Structures  4.0 SH  [ ]
In-depth coverage of programming concepts, with emphasis on data structures, including arrays,
structs, class objects, stacks, queues, and lists. Algorithms and complexity analysis for searching and
maintaining sorted and unsorted lists. Explicit pointer data type and dynamic allocation of memory.
Introduction to object-oriented programming and recursive functions.
**Prerequisite:** CSCI 171 or CSCI 181.

CSCI 201  Programming Strategies  1.0 SH  [ ]
Open to students with at least two programming courses who would like to gain a knowledge of team-
oriented programming strategies that enable them to solve problems quickly and accurately. This
knowledge is useful in many fields of problem solving, but will also prepare them for the ACM Scholastic
Programming contest. The course emphasizes fast analysis, design, and coding. May be taken up to
two times.

CSCI 220  Programming Languages  4.0 SH  [ ]
A survey and comparison of programming languages, learned through study and considerable
programming practice. Intended to form the programming language basis for courses in object-oriented
programming and database management systems.
**Prerequisite:** CSCI 182.

CSCI 240  Object-Oriented Programming  4.0 SH  [ ]
Students design, code, document, and debug computer programs using classes, object-oriented
design, inheritance, polymorphism, and recursion. They demonstrate problem-solving skills using
object-oriented concepts.
**Prerequisite:** CSCI 220.

CSCI 263  Computer Software Architecture  3.0 SH  [ ]
Topics include assembly language programming and ISA design issues, low-level OS concepts
including memory management, cache, I/O and process/thread management.
**Prerequisite:** ENGR 262.

CSCI 310  Software Engineering  3.0 SH  [ ]
Methodologies for developing and maintaining large complex software systems. Case studies and
team exercises are included. Focus on complete software life cycle, including specification, design,
implementation, testing, debugging, and documentation.
**Prerequisite:** CSCI 171 or CSCI 181.
**Class Level Restriction:** Junior and Senior only.

CSCI 320  Database Management Systems  4.0 SH  [ ]
The utilization of database systems within organizational settings. Includes features of database
management systems, various data models, database design, normalization, SQL, query optimization,
various indexing techniques, and data integrity and recovery issues.
**Prerequisite:** BUAD 215 or CSCI 240.
**Class Level Restriction:** Junior and Senior only.
CSCI 324  Web Development  4.0 SH
Design and development of client and server-side web applications, using libraries, and technologies relevant as of the time of course offering. Coverage of scripting language concepts including closure, dynamic typing, and prototypes. The class is project-oriented, and students will complete a significant web application.
Prerequisite: CSCI 320.
Class Level Restriction: Junior and Senior only.

CSCI 330  Operating Systems  3.0 SH
The inner workings of several popular multiprogramming operating systems are compared. Topics include processor scheduling, device management, memory management, and parallel processing.
Prerequisite: CSCI 263.
Class Level Restriction: Junior and Senior only.

CSCI 333  Computer Networking  3.0 SH
Introduction to Local Area and Wide Area computer networks. Topics include network topologies, ISO protocols, Ethernet, TCP/IP, and network support for distributed computing. The course emphasizes the science of networking (as opposed to network administration).
Prerequisite: CSCI 171 or CSCI 181.
Class Level Restriction: Junior and Senior only.

CSCI 340  Advanced Algorithms  4.0 SH
Survey of algorithms, with an emphasis on the application of algorithmic theory to the solution of practical problems. Categories include algorithm taxonomy, greedy algorithms, dynamic programming, divide-and-conquer, and intelligent search. Various specific algorithms exemplifying these categories are drawn from a variety of application areas such as compression, encryption, graph problems, sorting, computational geometry, and string matching.
Prerequisite: CSCI 240.
Class Level Restriction: Junior and Senior only.

CSCI 380  Compilers  4.0 SH
Concepts necessary for designing and developing a compiler. Topics include lexical analysis, parsing, semantic analysis, symbol table management, and code generation. Students will implement a small compiler as a project.
Prerequisite: CSCI 240 and CSCI 263.
Class Level Restriction: Junior and Senior only.

CSCI 390  Topics in Computer Science  1.0-5.0 SH
Topics will vary from year to year based on the needs and interests of students and instructor. All offerings will involve in-depth study of a specific topic, software design, the construction of working software, and self-instruction in addition to standard instruction. May be taken more than once if topics are different.
Prerequisite: CSCI 240.
Class Level Restriction: Junior and Senior only.

CSCI 410  Capstone Proposal  1.0 SH
Students research and prepare a capstone proposal in preparation for completing a computer science capstone project. The proposal must be approved by the computer science faculty prior to taking CSCI 415.
Class Level Restriction: Junior and Senior only.

CSCI 415  Capstone  3.0 SH
A student-designed project with a significant programming component which pulls together the student's major. Choices include independents, internships, or group projects.
Prerequisite: CSCI 410.
Class Level Restriction: Junior and Senior only.
CSCI 421  Computer Issues Seminar  3.0 SH  [  ]
An examination and discussion of pressing computer issues facing society. Topics include personal impact, societal impact, artificial intelligence, privacy, and ethics. Majors from other disciplines are encouraged to take this course if they have taken any other computer science course.

Class Level Restriction: Senior only.