



CSIS-111B: FUNDAMENTALS OF COMPUTER PROGRAMMING

Effective Term

Fall 2025

BOT Approval Date

01/16/2025

Discipline

CSIS - Computer Sci/Info Systems

Course Number

111B

Course Title

Fundamentals of Computer Programming

Short Title

Computer Programming

Credit Status (CB 04)

D - Credit - Degree applicable

Units**Lecture Units**

3.00

Total Units

3.00

Hours**Lecture Contact Hours**

48-54

Out of Class Hours Lecture

96-108

Total Contact Hours

48 - 54

Total Out of Class Hours

96 - 108

Total Student Learning Hours

144 - 162

Distance Education

Yes

Distance Education**Distance Education Type**

Both Fully Online and Hybrid Online

Fully Online Delivery Requirements:

- a. Students must be notified via the college schedule of classes and the syllabus for the class if proctored tests are required for this course
- b. Any planned face-to-face meetings, such as an orientation or study session, must be optional

c. The MSJC curriculum committee requires the use of accessible, asynchronous discussion as a component of every fully online course

Are you using publisher content?

No

Regular Substantive Interaction

Instructor-to-student, student-to-student, and student-to-content contact shall be distributed in a manner that will ensure regular substantive interaction (RSI) is maintained not only weekly but also for the duration of the course. Instructor interactions should be equivalent in time, quality and consistency to the engagement students would receive in a face-to-face course—for example, through announcements, discussion participation, personalized feedback, video messages, or other instructional communications. Instructors are expected to be responsive to student inquiries within a 24-72 hour time frame excluding holidays. RSI will be maintained through a variety of methods:

- Orientation - Students must be oriented to the online and face-to-face portions at the start of the course.
- Announcements - Announcements using the course management system must be posted at least weekly to keep students current on course events, due dates, materials, etc.
- Discussion Forums - Content-related discussion forums within the designated CMS will include appropriate instructor participation and will be initiated and maintained, with timely feedback provided. Instructors are required to monitor the discussions and related activities to provide guidance and direction. Open-ended discussion boards may be used to further learning and interaction.
- MSJC Communication – Instructor will use MSJC-administered communication tools (such as email or the designated CMS communication tools) to regularly communicate with and provide ease of access to students.
- Timely Feedback - Timely feedback on student work will be provided by the instructor, which may include comments and/or rubrics, within the Grades area of the designated CMS.
- Scheduled Meetings – Hybrid and synchronous courses will meet at regularly scheduled times.

ADA Compliance - All course interactions and content must be designed with accessibility in mind and meet ADA requirements.

C-ID (Admin Only)

C-ID ITIS 130

Catalog Description

This course will introduce students with no prior programming experience to the fundamentals of computer programming. These are foundation concepts for nearly all modern programming languages including Visual Basic, C++, C# and Java. Topics include sequence, repetition, and selection control structures. Advance topics include arrays, file I/O, and an introduction to the principles of object-oriented programming. One or more high-level programming languages will be used to reinforce the general concepts presented in this course.

Codes

TOP Code (CB 03)

0707.10* - *Computer Programming

CIP Code

11.0201 - Computer Programming/Programmer, General.

Repeatability Code

No

Grading Option

Letter grade OR P/NP

GE Information

Check GE Types Requested

AA/AS D2 - Communication & Analytical Thinking

Learning Objectives

Learning Objectives

Learning Objective

1. Plan a business solution which incorporates each step of the program development cycle.
2. Compose simple algorithms, flowcharts, and pseudo-code.
3. Contrast and compare high-level programming languages as to suitability to a task.

4. Develop business solutions which demonstrate the proper use of programming syntax including the effective and appropriate use of variables, expressions, and functions.
5. Apply appropriate Sequential, Repetition, and Selection control structures to solve specific programming problems.
6. Develop business solutions which demonstrate the role of arrays and File I/O through language specific application of these techniques.
7. Review and and reflect on the contributions that have been made to computer sciences from marginalized groups such as women and people of color.

Learning Outcome Information

Course Learning Outcomes

Learning Outcome
1. Students will develop a program that correctly implements the for, while, and do while looping structure.
2. Students will code and implement the if, if else, conditional, and switch decision constructs.
3. Students will develop algorithms that implement arrays for storage and data retrieval.
4. Students will demonstrate the use of functions for modular programming.
5. Students will code a complex (operations do not share the same level of precedence) algebraic expression in the target language.
6. Students will learn the benefits of working in diverse and inclusive environments.

Program Learning Outcomes

Learning Outcome
1. Recognize that a system consists of people, procedures, hardware, software, and data within a global environment.
2. Apply systems concepts in the investigation, evaluation, and resolution of information technology problems.
3. Recognize how the very large amounts of data collected by modern organizations can be used to review, redesign, and improve processes.
4. Employ applications software and software tools in the application of information technologies to help individuals, groups, and organizations achieve their goals.
5. Analyze existing processes based on interviewing, observation, documentation, analysis and other similar methods.
6. Research and apply industry reference models and best practices in order to improve process designs.
7. Assess, manage, and control IT risks.
8. Demonstrate working effectively as a member of a team to accomplish common goals.
9. Analyze technical information, as well as listen effectively to, communicate orally with, and prepare memos, reports and documentation for a wide range of audiences.
10. Investigate and assess new sources of information and learning opportunities to stay abreast of emerging information and computing technologies.
11. List career paths related to the program of study, as well as any qualifications and/or professional certifications that may be associated with those careers.

Institutional Learning Outcomes

Aesthetic Awareness: The student will identify and evaluate how aesthetic expression leads to an appreciation of social and cultural context as well as the values of art and nature.

Communication: The student will effectively express and exchange ideas through listening, speaking, reading, writing, visual works, and other modes of interpersonal expression.

Critical Thinking: The student will analyze problems, gather and synthesize relevant information, evaluate ideas, information, and evaluate alternative points of view to create, innovate and implement effective solutions.

Cultural Awareness and Humility: The student will demonstrate knowledge of and sensitivity toward individuals of diverse ethnicity, age, gender, sexual orientation, and religious affiliations, as well as toward those individuals with diverse abilities and socio-economic classes to respectfully interact with individuals of diverse perspectives, beliefs and values while being mindful of the limitation of their own cultural frameworks.

Information and Technology Literacy: The student will access, interpret, evaluate, and apply relevant information sources and digital media effectively, and in an ethical and legal manner.

Social Awareness: The student will recognize and analyze the interconnectedness of global, national, and local concerns, analyzing cultural, political, social, and environmental issues from multiple perspectives and appreciate similarities and differences among cultures.

Civic
Personal
and Professional Responsibility

Content

Course Lecture Content

1. The Stored Program Concept
 - a. History of Programming
 - i. Katherine Goble Johnson, Dorothy Vaughan, and Mary Jackson, who worked as "human computers" at NASA during the Space Race.
 - ii. Ethics and Bias in Technology
 - b. Compilers and Interpreters
 - c. Procedural vs. Object-Oriented Programming
 - d. High-level vs. Low-level Languages
2. Structured Design Concepts
 - a. Binary Encoding
 - b. System Development Life Cycle
 - c. Software Testing
 - d. Clean Coding
3. Anatomy of a Computer Program
 - a. Algorithms
 - b. Flowcharting
 - c. Pseudocode
 - d. Decision Tables
4. Simple Data Types and Operators and Expressions
 - a. Constants
 - b. Variables
 - c. Data Types
5. Arithmetic, Boolean, and Relational Operators Expressions
 - a. Numeric Expressions
 - b. String Expressions
6. System Defined Functions
7. Input Assignment and Processing Output
8. Control Structures
 - a. Sequence Control
 - b. Repetition Control
 - i. Counter Control Loops
 - ii. Query Control Loops
 - iii. Sentinel Control Loops
 - iv. Nested Loops
 - v. Selection Control
 - c. Decision Structures
 - i. If Structures
 - ii. If-Else Structures
 - iii. Nested If-Else Structures
9. Multi-way Selections
10. Arrays
 - a. Declaration and Initialization
 - b. Input of Data into Arrays
 - c. Output of Data from Arrays
 - d. Operations on Arrays (Searching and Sorting)
11. File I/O
 - a. Sequential Files
 - b. Using Try/Catch/Finally with file I/O
12. Modular Programming

a. User defined Functions and Procedures

13. API's, Libraries, and Frameworks

Methods of Instruction

Method

Individual Projects

Integration

Guided Practice provides a context for demonstrating language syntax, features, and usage. Guided Practice is facilitated through screen projection or other technology mediation. A key feature of Guided Practice is the student's ability to practice or follow along with their own computer. Appropriate Guided Practice topics include the appropriate and effective use of language keywords such as PRINT and INPUT, control structures, and the appropriate use of variables, expressions, and functions.

DE Adaptations for MOI

Accessible guided practice videos demonstrating various computer programming concepts with source code will be provided to students via the cms. Students will follow along on their personal computer which has a source code editor already installed (students are taught how to install the software onto their computer during the first week of class using accessible text-based or video demonstration instructions distributed via the course shell on the college CMS). Students will post their source code to the college CMS Assignments by the stated due date and the instructor will evaluate and score the students' work using a rubric which is posted with the CMS assignment.

Method

Lecture

Integration

Lectures include contributions and achievements of diverse programmers and computer scientists from diverse backgrounds highlighting pioneers and contemporary figures of different races, ethnicities, and sexual orientations to demonstrate that the field is inclusive to all. Lectures elaborate on programming syntax and other language features. Lecture also elaborates considerations such as procedural vs. object oriented programming and the difference between compilers and interpreters in choosing a programming language or development tool for a particular task.

DE Adaptations for MOI

Online lectures elaborate programming syntax and other language features and will be available to students asynchronously via the CMS and may include accessible Camtasia style videos (screen capture), audio podcasts (mp3 or similar), written or illustrated content (.pdf), and may be enhanced by accessible Wikis, Blogs, and other web-based technologies. Online Lectures may also be enhanced or elaborated on using other accessible online resources such as Youtube, and Open Course content (i.e. Coursera, MIT OpenCourseWare, iTunes U, etc.). Asynchronous lecture content may be supplemented by live-session reviews (Zoom or similar).

Method

Reading

Integration

Reading from accessible online resources which elaborate on lectures, and inform guided practice on topics such as the appropriate use of constants, variables, and expressions. Readings may also include accessible formal language documentation sources.

DE Adaptations for MOI

All reading materials presented in this class will be accessible and provided within the CMS or links to other accessible web-based sources, e-Books, etc.

Method

Discussion

Integration

Classroom discussions based on lectures and videos presented by the instructor will introduce the benefits of working in diverse and inclusive environments, ethical issues in programming, such as algorithmic bias and the impact of technology on marginalized

communities. Students will be shown how their work can inadvertently perpetuate biases if not carefully considered. Students will be asked to interact with each other in a robust discussion forum.

DE Adaptations for MOI

Lectures and videos will be made accessible and available on the MSJC CMS. Online discussions will be implemented using discussion forums on the CMS.

Methods of Evaluation**Method**

Exams/Tests

Integration

Tests and Exams cover language syntax and conceptual concepts covered in the course including data types, relational operators, and expressions. In addition to course theory, Exams may include a programming or production component. Production or programming exam elements should be evaluated on a rubric that includes required execution, design, and style elements.

DE Adaptations for MOE

Students will access Tests and Exams using the course shell on the college's CMS Assignments feature. The instructor will score the tests and exams using a rubric provided specifically for the test or exam in the course shell of the college CMS.

Method

Projects

Integration

Each Programming project will require programming a solution to a programming problem through the application of techniques and theory discussed in accessible lectures, and/or demonstrated in class guided practice sessions. Programming Projects integrate multiple topic covered in the course, including at least one cumulative project that integrates sequence, repetition, and selection control structures. Projects will be evaluated on a rubric that includes required syntax, design, and style elements.

DE Adaptations for MOE

Accessible guided practice videos demonstrating various computer programming concepts with source code will be provided to students via the course shell located on the college CMS. Students will follow along on their personal computer which has a source code editor already installed (students are taught how to install the software onto their computer during the first week of class using accessible text-based or video demonstration instructions distributed via the course shell on the college CMS). Students will post their source code to the college CMS Assignments by the stated due date and the instructor will evaluate and score the students' work using a rubric which is posted with the CMS assignment.

Method

Written Assignments

Integration

Written Assignments will demonstrate the appropriate application of course theory in problem solving including the appropriate use variables, constants, expressions, and functions. Written Assignments will require tracing of program execution and the identification of syntax and semantic errors. Trace Problems will be evaluated on whether a given answer matches program execution. Debug problems should be evaluated on the extent to which specific syntax and semantic errors are identified, and suggested changes resolve or correct a given error.

DE Adaptations for MOE

Written Assignments will demonstrate the appropriate application of course theory in problem solving including the appropriate use variables, constants, expressions, and functions. Written Assignments will be managed through the CMS (.pdf files in conjunction with CMS Assignment links). Written Assignments will require tracing of program execution and the identification of syntax and semantic errors. Trace Problems will be evaluated on whether a given answer matches program execution. Debug problems should be evaluated on the extent to which specific syntax and semantic errors are identified, and suggested changes resolve or correct a given error.

Method

Quizzes

Integration

Students will complete formative quizzes weekly based on the topics presented in the weekly instructional material. Students will be given the option of repeating the quizzes as many times as they want to become more familiar with the material and to provide meta-cognitive evaluation to determine how well they are retaining the instructional information and to prepare for the mid-term and final exams.

DE Adaptations for MOE

Students will complete formative quizzes weekly provided in the CMS course shell based on the topics presented in the weekly instructional material. Students will be given the option of repeating the quizzes as many times as they want to become more familiar with the material and to provide meta-cognitive evaluation to determine how well they are retaining the instructional information and to prepare for the mid-term and final exams.

Assignments**Assignment Type**

Writing

In-class and/or outside of class

Outside of class

Formative or Summative

Summative

Assignment

Write a computer program that takes input, processes the input, and generates some type of output based on provided specifications posted on the assignment page in the cms course shell. Then copy and paste your source code to the assignment page found in the cms course shell by the due date listed in the syllabus. The instructor will score the assignment no later than 2 days after the due date listed in the syllabus. Your score will be based on whether or not the assignment was submitted on-time, submitted in the correct format, your program meets the assignment specifications, program builds without failure and runs without error. You will then have the opportunity to improve your score by making corrections to your source code based on the instructor's comments.

How assignment will be adapted in online format

Students will post their assignment to the course shell in the CMS. The instructor will score the assignments based on how well the student's program meets the criteria listed above and identify any errors by line number in the comments section of the grading page.

Assignment Type

Writing

In-class and/or outside of class

Outside of class

Formative or Summative

Formative

Assignment

After viewing this week's lecture, post a journal entry to this week's journal page in the CMS identifying the key points that you learned from this week's instructional materials. This journal can be useful for your own reference and also help you to better retain the information you are learning in this class. You will be scored based on accuracy and completeness.

How assignment will be adapted in online format

The journals are recorded and scored in an assignment module of the cms course shell, so no changes are needed for the online version of this class.



Assignment Type

Other

In-class and/or outside of class

In-class

Formative or Summative

Formative

Assignment

After viewing the lecture on the Anatomy of a computer program. Think of a program you would like to create. Using the specified assignment page on the course shell of the college cms write out, step-by-step, what the input for your program would be, what the processing steps of your program would be, and what the output of your program would be. You will be graded on how well you identify the proper inputs, processes, and outputs for your program.

How assignment will be adapted in online format

Students will access the lecture from the course shell on the cms. They will then post their algorithm to the specified assignment page in the cms. The instructor will evaluate their assignment based on how well the student identified the proper inputs, processes, and outputs for their program.

Assignment Type

Reading

In-class and/or outside of class

Outside of class

Formative or Summative

Formative

Assignment

Read the online instructional content about the array data type. During your reading pay close attention to how to declare an array, populate an array, manipulate array values, and output the values in an array. Write a program that creates a 10 element array, populates it with randomly generated integer values, sorts the array contents by numerical value, and then prints the contents of the array to the screen. You will be evaluated based on whether your program compiles, runs without error, and meets the program requirements described above.

How assignment will be adapted in online format

All instructional materials in this class will be accessible and made available via the CMS; so no changes are necessary for the fully online or hybrid versions of this course. Scoring will also occur using the cms grading page for the assignment.

Assignment Type

Other

In-class and/or outside of class

In-class

Outside of class

Formative or Summative

Formative

Assignment

After viewing the movie Hidden Figures, discuss how Katherine Goble Johnson, Dorothy Vaughan, and Mary Jackson, female African-American mathematicians (also known as the first computers), played a pivotal role in John Glenn's launch into orbit. Your discussion must also include how they had to deal with racial and gender discrimination at work. You will be graded on completeness, accuracy, and insightfulness.

How assignment will be adapted in online format

For the online adaptation, students will access the movie on YouTube where it is offered free of charge. For the discussion portion the college CMS will be used to create a discussion forum which students will be required to post to and interact with other student posts. Students will be scored based on their participation in the discussion forum.

Course Materials

Textbooks

Reed, M (2024). C# & C++: 5 Books in 1 - The #1 Coding Course from Beginner to Advanced (optional)
ISBN: 978-1549724541

Class Size Information

Class Size

35

Class Size Category

B - Facilitated learning 35

Diversity and Inclusion

Explain how diversity and inclusion(e.g., gender, culture/race, sexuality, etc.) is infused into the course.

Inclusion will be incorporated in the course using several lectures which explain the importance of inclusive design and accessibility, including the following:

1. Relevance: Inclusive design is a foundational concept that emphasizes the importance of building computer applications that are accessible to all users, regardless of their abilities or backgrounds.
2. Technical Integration: It ties directly into the practical aspects of application development, such as using projects that encourage students to develop solutions to societal issues, such as accessibility apps, tools for underrepresented communities, or programs that address gender disparities in tech..
3. Awareness: It introduces students to the social responsibility of application developers to create equitable digital experiences. They will learn that good application design considers diverse user needs, including those with disabilities.
4. Skill Development: Understanding and implementing accessibility features is a valuable skill in the tech industry. This knowledge can set students apart as developers who prioritize user experience for everyone.

Describe how assignments, instruction, evaluation, course content, and interactions are contextualized for diverse learners promoting an equitable course.

Multiple teaching modalities, such as visual aids, audio resources, and hands-on activities that cater to different learning styles, will be incorporated into the lectures and written material for the class. Assignments will be framed around real-world application development projects with each offering hands-on experiences with how to build accessible applications from clearly explained instructive coding models. Students will be encouraged to share their experiences related to inclusive application design and accessibility concepts in an inclusive discussion environment so that students will feel comfortable expressing themselves and engaging in conversation.

The course lecture material will be infused with diversity. It will reflect diverse voices and perspectives on the field of application development. The works of researchers, practitioners, and innovators from various backgrounds and cultures will be highlighted. Rubrics describing exactly how the students will be scored will be provided for each assignment.

Interactions

Explain how the course will incorporate student-to-student interaction.

The course will include facilitated weekly discussions so students can discuss the field of application development, how the field benefits from accommodating diverse and inclusive environments, what is algorithmic bias and how to detect it, accessibility concepts, share insights, and ask questions. Questions will be posed, and case studies will be offered to stimulate discussion. Within the observation and demonstration aspect of the class, students will be encouraged to share their ideas and thoughts to solve accessibility problems and content biases. For hybrid and online classes the cms discussion forums will be used to pose the questions/case studies and allow students to post their thoughts and interact with other students by responding to their initial posts.

Explain how the course will incorporate instructor-to-student interaction.

Instructor-to-student interaction will be incorporated through regular lectures, feedback, and open communication channels, which include question-and-answer sessions where student questions can be answered and clarified. Tutors, who have previously completed the CSIS 111B course, will be made available for peer tutoring. Office hours will be provided where students can meet with the instructor in one-on-one or small groups to discuss course content, seek clarification, or receive additional help and feedback on assignments. Emails and post in Canvas messaging to the instructor will be promptly replied to (within 48 hours). Personalized feedback on assignments will be provided weekly. For hybrid and online classes, email and CMS messaging will be used for office hours. CMS discussion forums, assignments pages, and grading comments will also be used for instructor-to-student interaction.

Explain how the course will incorporate student-to-content interaction.

The lectures emphasize interaction related to data processing, incorporating small coding problems, formative quizzes composed of questions about algorithmic structures, and open-ended discussions which also incorporate data processing topics. At the end of major milestones in the course, formative quizzes will be given, allowing the students to gauge their progress. Formative quizzes will also be placed within the reading material so students can test their understanding of the material being presented. Students will also be given instructive feedback on their assignments with an opportunity to improve their assignment based on the feedback given. For hybrid and online classes the CMS assignments will be used to distribute text-based and/or video-based instructional content. The CMS quizzes will be used for both formative and summative assessments.

Explain how the course will incorporate student-to-self interaction (metacognitive).

The lecture will include exercises that prompt students to think about how they best learn data processing topics, i.e. given sample code and asked to modify it to be more efficient. Students will be encouraged to reflect on their study habits, identify effective data processing learning strategies, and suggest potential adjustments.

Assignments that involve best practice for implementing accessibility and preventing bias analysis will include metacognitive reflection prompts at the end of the assignment. An example is "How confident are you in your solution and why?"

For hybrid and online classes practice quizzes and ungraded surveys will be used to provide meta-cognitive feedback as to how well the student is progressing in the class.

Key: 1019