



CSIS-113C: C# PROGRAMMING - LEVEL 1

Effective Term

Fall 2023

BOT Approval Date

10/13/2022

Discipline

CSIS - Computer Sci/Info Systems

Course Number

113C

Course Title

C# Programming - Level 1

Short Title

C# 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.

Catalog Description

This course introduces the principles of object-oriented programming using the C# programming language. Students will investigate and evaluate various programming design methodologies and apply them to C# programming problems. C# features that will be covered include language syntax, class definitions, control structures, method definitions, and basic data structures. No prior programming experience required.

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

Learning Objectives

Learning Objectives

Learning Objective

1. Contrast and compare major elements of the .NET Framework and explain how C# fits into the .NET platform.
2. Analyze the basic structure of a C# application and be able to document, debug, compile, and run a simple application.
3. Declare, name, and assign values of a specific data type to variables.
4. Compose common statements to implement flow control using decision structures, looping, and exception handling.
5. Create methods that can return values and take parameters.
6. Declare, initialize, and use arrays to solve data storage and retrieval problems.
7. Explain the basic concepts and terminology of object-oriented programming.
8. Construct programs that demonstrate the use of common objects and reference types.
9. Create common GUI components (Windows Forms) and modify their properties at both run time and design time.

10. Identify and assess the benefits of diversity, equity and inclusion in the computer programming workplace (e.g. teamwork, creativity, innovation, competitive advantage, etc.).

11. Examine racial and gender biases in the development of artificial intelligence (AI).

12. Analyze the racial and gender gaps that exists in computer programming workplaces.

Learning Outcome Information

Course Learning Outcomes

Learning Outcome

Prepare a program / algorithmic design for a simple straight-line problem.

Code and implement a Do/while loop constructs for a given problem.

Code and implement a While loop construct for a given problem.

Code and implement a For loop construct for a given problem.

Create a robust test suite in support of a C# program design.

Implement a test module as part of the program design that will exercise a test suite

Identify issues regarding diversity, inclusiveness, and equity in computer programming jobs, the workplace, and in computer programming applications.

Program Learning Outcomes

Learning Outcome

Recognize that a system consists of people, procedures, hardware, software, and data within a global environment.

Apply systems concepts in the investigation, evaluation, and resolution of information technology problems.

Recognize how the very large amounts of data collected by modern organizations can be used to review, redesign, and improve processes.

Employ applications software and software tools in the application of information technologies to help individuals, groups, and organizations achieve their goals.

Analyze existing processes based on interviewing, observation, documentation, analysis and other similar methods.

Research and apply industry reference models and best practices in order to improve process designs.

Analyze technical information, as well as listen effectively to, communicate orally with, and prepare memos, reports and documentation for a wide range of audiences.

Investigate and assess new sources of information and learning opportunities to stay abreast of emerging information and computing technologies.

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. C# Basics
 - a. Data Types
 - b. Operators and Expressions
 - c. Introduction to Objects
2. Control Structures
 - a. Relational Operators
 - b. If / If else statements
 - c. While Loops
 - d. For Loops
 - e. Switch Statements
 - f. Do while loops
 - g. Break and Continue Statements
3. Methods
 - a. Built in Methods
 - b. Programmer-defined Methods
 - c. Passing Parameters by Reference and Value
 - d. Scope Rules
 - e. Recursion
 - f. Overloading
4. Arrays
 - a. Creating and Allocating Arrays
 - b. Built-in Array Operations
 - c. Arrays as Arguments
 - d. Multi-dimensional Arrays
5. Object Based Programming
 - a. Introduction to Classes
 - i. Variables and Method Scope
 - ii. Constructors
 - iii. Overloaded Constructors
 - b. Exception Handling
 - i. Exception Fundamentals
 1. Re-Throwing Exceptions
 2. Custom Exception Classes
6. Windows Forms
 - a. Main Form
 - i. Properties
 - ii. Labels
 - iii. Buttons
 - iv. Textboxes
 - v. Group Boxes
 - vi. Panels
 - vii. Tab Control
 - viii. Checkboxes
 - ix. Radio Buttons
 - x. Picture Boxes
 - xi. Link Labels
 - xii. List boxes
 - xiii. Checked List Boxes
 - xiv. Combo Boxes
7. Certification
 - a. MCSD Certifications
 - b. Preparing for the Test
8. Examining Computer Programming Jobs and the Computer Programming Workplace

- a. Diversity, Inclusion, and Equity Issues Related to Computer Programming Jobs and the Workplace
 - i. How the Digital Divide Impacts Computer Programming Applicant's Skill Levels
 - ii. Racial and Gender Biases in the Development of Artificial Intelligence
 - iii. Racial and Gender Gaps in the Computer Programming Workplace
- b. Where to Look for Jobs
- c. Professional Resumes
- d. Soft Skills
- e. Hiring Exams

Methods of Instruction

Method

Discussion

Integration

Weekly discussions will be conducted as a means to enhance lectures and discuss subjects listed in the course outline, such as how the digital divide impacts computer programming skill levels, the existence of racial and gender pay gaps in programming jobs, how to recognize gender and cultural biases in the development of artificial intelligence, comparing and contrasting the use and implementation of the for, while, and do while looping constructs, proper use of the if, if else, conditional, and switch decision constructs, increasing the efficiency of algorithms that implement arrays for storage and data retrieval, the correct use of primitives to create abstract data types, etc.

DE Adaptations for MOI

Student discussion will be achieved in online format using asynchronous threaded discussions in the accessible MSJC designated CMS course shell. Students will post and comment with other students and the instructor within the CMS discussion forums, or by participating in synchronous discussions using accessible video conferencing platforms. Students can also be divided into breakout groups during synchronous group meetings to compare their ideas before being brought back to the main session to compare their ideas as a class.

Method

Lecture

Integration

Lecture, with supporting accessible visual materials (slides presentation or multimedia) will introduce conceptual and practical skills such as how the digital divide impacts computer programming skill levels, the existence of racial and gender pay gaps in programming jobs, how to recognize gender and cultural biases in the development of artificial intelligence, coding and implementing for, while, and do while constructs; implementing if, if else, conditional, and switch decision constructs; the development of algorithms that implement arrays for data storage and retrieval; demonstrate the use of primitives to develop complex types; develop algorithms that show the appropriate creation and use of abstract data types; investigate racial and gender biases in the development of artificial intelligence.

DE Adaptations for MOI

Lecture content consisting of accessible Web-based multimedia presentations including, HTML generated content, source code demonstrations using video capture technologies, narrated slides presentations will be disseminated via the accessible MSJC designated CMS course shell. In addition synchronous lectures using accessible video conferencing platforms may be used.

Method

Observation and Demonstration

Integration

Guided practice and demonstration will illustrate lecture principles and reading assignments with a focus on how the digital divide impacts computer programming skill levels, the existence of racial and gender pay gaps in programming jobs, how to recognize gender and cultural biases in the development of artificial intelligence, fundamental C# programming concepts and applying them to problem solving and potential solutions to homework assignments in addition to discussing the gender gap in programming and programmers. Guided practice will include coding and implementing for, while, and do while constructs; implementing if, if else, conditional, and switch decision constructs; the development of algorithms that implement arrays for data storage and retrieval;

demonstrate the use of primitives to develop complex types; the development of algorithms that show the appropriate creation and use of abstract data types.

DE Adaptations for MOI

Student access to the guided practice and demonstrations will be accomplished by using accessible Web-based step-by-step written instructions along with accessible step-by-step demonstration videos deployed using the accessible MSJC designated CMS course shell. Hands-on activities will be similarly accommodated and supplemented with the use of software simulations and source code editors provided by the accessible MSJC designated CMS course shell.

Method

Directed Study

Integration

Web-based instructional content, including accessible pre-recorded video instruction, or in-class instructions when possible, will guide students step-by-step through writing source code that solve common programming problems utilizing recognized design patterns and incorporating SOLID principles. Examples would include using for loops to parse the content of arrays, using decision structures for providing flow control, examining a user interface to ensure diversity, inclusion, and equity. Students will post their source code using a link provided in the accessible MSJC provided CMS course shell. Students will be evaluated by the instructor using the same rubric the students were provided at the beginning of the directed study exercise. The instructor will identify any necessary corrections needed and students will be given an opportunity to make the appropriate corrections in order to improve their score.

DE Adaptations for MOI

Instructional content and scoring rubric will be linked to or provided directly in the accessible MSJC CMS course shell. Students will post their source code using the Assignments tool in the accessible MSJC CMS course shell. The instructor will score assignments and leave comments to students using the Grading tool in the accessible MSJC CMS course shell within one week of the assignment's due date.

Methods of Evaluation

Method

Exams/Tests

Integration

A midterm and a final exam requiring the student to write source code will be used for a summative evaluation of the student's understanding of the basic structure of a C# program, of creating, naming, and assigning values to variables; of how to compose statements for implementing flow control, looping, and exception handling; of how to create methods which accept parameters and return values, of how to create, initialize, and implement arrays, of the basic concepts of object-oriented programming, of how to create common GUI components and modify their properties at runtime as well as dynamically while a program is being interacted with by a user; to identify options and generates alternative approaches to a computing problem, and how to engage in multiple and simultaneous outcomes and activities, will be submitted via the assignment tool of the accessible MSJC designated CMS course shell.

DE Adaptations for MOE

Students will write their source code using a source code editor on their own digital device of choice or by accessing the accessible MSJC remote desktop system which provides online source code editors for student use. Students will submit their completed source code using the accessible MSJC designated CMS course shell's Assignment tool which also provides an accessible scoring rubric for student viewing. Scoring and commenting by the instructor will be completed using the CMS Grading tool within one week of the assignment's due date.

Method

Projects

Integration

Student will be asked to complete projects consisting of writing and discussing how the digital divide impacts computer programming skill levels, the existence of racial and gender pay gaps in programming jobs, how to recognize gender and cultural

biases in the development of artificial intelligence. Students will also be submitting source code projects that include C# syntax and logic problems; looping structures (for, while, do-while); decision structures (if, if else, switch, conditional); the use of arrays to solve storage and retrieval problems; the development of algorithms that demonstrate the proper use primitives to create complex types; the development and proper use of abstract data type problems will be required. Guided practice/discussion projects are intended to be formative assessments and will be evaluated based on the program running to completion without error or warning; how well directions are followed; program correctness; proper input and output. Student will receive instructor feedback within one week of the assignment due date. After reviewing the instructor's feedback on a project, the student will have the opportunity to make any corrections necessary in order to achieve a higher score.

DE Adaptations for MOE

Students will submit project assignments via the assignment and/or discussion forum tool of the accessible MSJC designated CMS course shell. Feedback to students along with evaluation will be given to the student via the accessible MSJC designated CMS course shell. Students will receive instructor feedback within one week of the assignment due date.

Method

Homework

Integration

Weekly programming assignments will apply important concepts and skills to the student's experience outside the classroom. The focus will be on subjects like how the digital divide impacts computer programming skill levels, the existence of racial and gender pay gaps in programming jobs, how to recognize gender and cultural biases in the development of artificial intelligence, the development and correct implementation of abstract data types, correct usage of the different looping constructs like the for, while and do-while loops, the implementation of if, else if, switch, and conditional decisions, racial and gender bias in the development of artificial intelligence and storage and data retrieval techniques using arrays. Depending on the weekly assignment type coding or discussion assignment will be evaluated based on a provided rubric accessed from the designated CMS assignment page and will include evaluating the student's adherence to the assignment specifications like proper use of grammar and spelling, how well they demonstrate their understanding of the instructional materials, how well directions are followed, timely submission of assignment, written programs running to completion without error or warning; written program's correctness; written program's proper implementation of input, processing, and output.

DE Adaptations for MOE

Students will complete and submit homework assignments using the accessible MSJC designated CMS course shell tools like the assignment tool, or the discussion forum tool which will include a student accessible scoring rubric. Scoring and commenting by the instructor will be done using the scoring rubric and submitted using the CMS grading tool and will be completed within one week of the homework's due date.

Assignments

Assignment Type

Writing

In-class and/or outside of class

In-class

Formative or Summative

Formative

Assignment

Write a simple calculator program using looping and decision structures. Your program should ask for three things, two whole numbers and an operator, in the form of an expression like: $3 * 2$. Use a select case structure to determine what operation needs to be performed on the two numbers. Your program should handle the arithmetic functions Add, Subtract, Multiply, and Divide (Depending on the operator entered). Watch out for division, meaning be sure to handle divide by zero errors. Use an if/else block to do the error checking. This check should only be done if the operation is a division operation. Also, you must have a default statement to handle when an invalid operator has been entered. Hint - Be careful on what you are switching on. Remember, literal characters need to be enclosed in single quotes 'a'.

Submit your assignment using the link provided in the accessible MSJC designated CMS course shell.

How assignment will be adapted in online format

The assignment will be created using the assignment editor found in the accessible MSJC designated CMS course shell. The assignment will be distributed and collected using the assignments tool of the CMS course shell. Scoring and instructor comments will be provided to students via the CMS course shell's built-in grading tool within one week of the assignment due date.

Assignment Type

Reading

In-class and/or outside of class

Outside of class

Formative or Summative

Formative

Assignment

Read the Data Types chapter in your textbook paying attention to the details of each data type. Meaning identify the data type name, the keyword used to represent the data type in source code, the size in bytes of the data type and whether it is a primitive or complex data type.

In the accessible MSJC designated CMS course shell locate the Discussion forum for Data Types and post your observations of the data types with suggestions as to when each data should be used in varying types of problem solving scenarios. After completing your post, read and reply to at least two other student post with a substantive evaluation of their post.

Your post will be evaluated for correctness of information based on a scale of 1 - 10, 1 being not very accurate, 5 being somewhat accurate, 8 being nearly accurate and 10 being perfect. You will also receive 5 points for each substantive reply you make on other student post; up to 10 points maximum.

How assignment will be adapted in online format

Students will access the instructional content using a pre-purchased hard copy of the textbook or the eBook that is provided within the CMS course shell. Students will post their initial post and replies to other student posts using the CMS discussion forum tool which will include a student accessible scoring rubric. Scoring and instructor comments will be provided to students via the CMS course shell's built-in grading tool within one week of the assignment due date.

Assignment Type

Writing

In-class and/or outside of class

In-class

Outside of class

Formative or Summative

Formative

Assignment

After reading/viewing your instructor's lecture material(s). Do some research and select a figure from a marginalized group who has had a major impact on the world of modern computer programming. Write two or three paragraphs which explains the impact this figure has had on the field of Computer Science and in your mind, what makes this person important. Please cite all sources used in your research. Post your completed document using the designated assignment link in the accessible MSJC designated CMS course shell.

Your instructor will score your assignment based on the provided rubric within one week of the due date of the assignment. After reviewing your instructor's comments you will be given the opportunity to make corrections and achieve a higher score.

How assignment will be adapted in online format

The lecture material(s) will be access by students using the designated Assignment page. Student's completed assignment(s) will be submitted via designated CMS Assignment page which will include a student accessible scoring rubric. Scoring and instructor comments will be provided to students via the CMS course shell's built-in grading tool within one week of the assignment due date.

Assignment Type

Other

In-class and/or outside of class

In-class

Outside of class

Formative or Summative

Formative

Assignment

After reading/viewing your instructor's lecture on identifying and assessing the benefits of diversity, equity, and inclusion in the computer programming workplace, open Module 3 in the accessible MSJC designated CMS course shell. Click the link to access assignment 3 and write a minimum 200 word post in the designated discussion forum consisting of what you learned from the lecture, what your observations are of the subject, and include any real-world experiences you might have had related to this subject. Then read at least two other student posts and post at least one thoughtful reply to that student's post following the rules of netiquette. Your assignment will be scored and commented on by your instructor based on the provided rubric using the course shell's grading tool within one week from the due date of the assignment as listed in the course syllabus. After reviewing your instructor's comments you will be given an opportunity to make corrections and improve your score.

How assignment will be adapted in online format

The instructor's lecture consisting of an accessible narrated slides presentation will be provided and accessed from the accessible MSJC designated CMS course shell. Student's initial 200 word post and replies to other student posts will be done using the instructor designated CMS discussion forum which will include a student accessible scoring rubric. Instructor scoring and commenting will be completed using the CMS grading tool within one week of the assignment's due date.

Assignment Type

Other

In-class and/or outside of class

In-class

Outside of class

Formative or Summative

Formative

Assignment

After reading/viewing the instructor's lecture explaining racial and gender biases in the development of artificial intelligence (AI), submit a 200 word or more post to the accessible MSJC designated CMS course shell's discussion forum consisting of what you learned from the lecture and what your thoughts or actual experiences regarding the subject are. Then read two other student posts and reply with at least one thoughtful response to each post that contributes something of value to the conversation. Your instructor will also be posting thoughtful replies and comments to your posts which you should reply to if prompted to do so. Your submissions will be scored and commented on by your instructor based on the provided rubric within one week of the assignment due date. After reviewing your instructor's comments you will be given the opportunity to make corrections and improve your score.

How assignment will be adapted in online format

The instructor's lecture consisting of accessible video, audio, and/or narrated slide presentation will be provided and accessed from the designated CMS Assignment page which will include a student accessible scoring rubric. Student's initial assignment submission and replies to other student posts will be completed using a designated CMS discussion forum. Instructor grading and commenting will be completed using the CMS grading tool within one week of the assignment's due date.

Assignment Type

Other

In-class and/or outside of class

In-class

Outside of class

**Formative or Summative**

Formative

Assignment

After reading/viewing the instructor's lecture materials analyzing the racial and gender gaps that exists in computer programming workplaces., submit a 200 word or more post to the accessible MSJC designated CMS course shell's discussion forum consisting of what you learned from the lecture and what your thoughts or actual experiences regarding the subject are. Then read two other student posts and reply with at least one thoughtful response to each post that contributes something of value to the conversation. Your instructor will also be posting thoughtful replies and comments to your posts which you should reply to if prompted to do so. Your submissions will be scored and commented on by your instructor based on the provided rubric within one week of the assignment due date using the CMS grading tool. After reviewing your instructor's comments you will be given the opportunity to make corrections and improve your score.

How assignment will be adapted in online format

The instructor's lecture consisting of accessible video, audio, and/or narrated slide presentation will be provided and accessed from the designated CMS Assignment page which will include a student accessible scoring rubric. Student's initial assignment submission and replies to other student posts will be completed using a designated CMS discussion forum. Instructor grading and commenting will be completed using the CMS grading tool within one week of the assignment's due date.

Assignment Type

Other

In-class and/or outside of class

In-class

Outside of class

Formative or Summative

Summative

Assignment

By the due date listed in the course syllabus complete the midterm exam. You will access the exam using the link provided in the accessible MSJC designated CMS course shell to the exam page. The exam consists of 50 true/false, multiple choice, and multiple answer questions related to the material you learned in modules 1 - 7. This exam is not timed but must be completed once it is started. Once completed you will be shown your score on the exam completion page and in your course gradebook under the midterm exam heading.

How assignment will be adapted in online format

Students will complete the exam using the designated CMS Assignment Quizzes tool. Questions will be randomly selected from a pool of questions derived from each of the module 1-7 quizzes and presented to the student by the Assignment Quizzes tool one at a time. The exam will be programmatically scored based on the student's submission to each question and the total score earned by the student will appear on the completion page of the Exam and in the CMS gradebook.

Course Materials**Textbooks**

Gaddis, Tony (2019). Starting Out With Visual C# Fifth Edition Pearson. ISBN: 9780135183519

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.

The course revision operates with a foundational intention of diversity, inclusion, and equity. Objectives have been updated to stress the necessity of teaching diverse content from a multicultural and intersectional perspective. They include identifying and assessing the benefits of diversity, equity, and inclusion in the computer programming workplace; examining racial and gender biases in the development of artificial intelligence, and analysis of the racial and gender gaps that exists in programming workplaces. This revision better articulates how the instruction and classroom community should offer opportunities to foster empathy, expose variant interpretations, and exhibit diverse societal concerns.

The instructional content, assessments, and assignments portions of the COR have also been revised to better align with our updated objectives to create opportunities for students of varying backgrounds and aptitudes to hone new skills while acknowledging and honoring their existing experiences and knowledge base. Instructional content has been chunked to provided scaffolded learning experiences. Lectures will be multimodal consisting of narrative, multimedia, in-class and online discussion forums, and interactive presentations to support learners' varying aptitudes for processing the subject matter. Assignments are primarily formative assessments and have also been revised to include diverse, inclusive, and equitable examples and problem sets. Weekly discussions are being used in addition to video, text-based, and classroom lecture to give student the ability to take ownership for their own learning experiences. For example, writing and discussing how the digital divide impacts society in addition to how it impacts their own lives. Assessments are primarily formative giving students the opportunity to not only improve their assessment scores, but also giving them an opportunity to be metacognitively aware of their understanding of the subject matter to help improve their scores on the summative assessments.

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

This course implements support for diverse learners by using active learning strategies like chunking instructional content into logical concise blocks of information and using numerous formative assessments that align with the revised course objectives and content to enhance the learning experience and promote increased success on summative assessments. For instance, in many assignments after submitting their work, students will receive comments from their instructor and if necessary will be given an opportunity to make changes and improve their scores. In addition, learning experiences are being enhanced by including in-class and out-of-class assignments and group discussions that explore content material through the lens of diversity, inclusiveness, and equity. In instructions, we have specified that discussions and class activities are spaces in which students can bring in their own cultural experiences through the application of computer programming concepts by allowing for alternative solutions to the problems posed. Discussions, group activities, and lectures have been revised to support the infusion of student feedback and interaction with the instructor. The DE portion of the course ensures that all instructional material is accessible and that all applications used in the course adhere to accessibility guidelines.

Interactions

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

For both face-to-face and online modalities, group work will give students the opportunity to collaborate on application of the .NET framework and platform, the analysis of basic C# computer programming structures, arrays for solving data storage and retrieval problems, options for generating alternative approaches, the examination of biases in the development of artificial intelligence and pay gaps in computer programming workplaces. In both modalities, students will be given frequent opportunities to engage in peer-to-peer discussions designed to foster community and encourage students to experience internet-based applications from different countries and of different socioeconomic backgrounds.

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

Using the course CMS discussion forums, breakout rooms in video-conferencing software, and/or in-class, the instructor will participate in class discussions asking students to apply critical analysis and/or theoretical lenses, clarify concepts, pose questions, and or/encourage students to make connections between course concepts and larger societal concerns, and examine computer programming systems and workplaces through an intersectional lens; explore socio-historical contexts, encourage students to make connections between course concepts and larger societal concerns, and examine common computer programming activities through an intersectional lens; will offer lectures on course concepts, critical approaches, variant interpretations, socio-historical contexts, and composition strategies.

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

In both face-to-face and online modalities, students will reference accessible materials such as textbooks, written lecture materials or transcripts of lectures, illustrative images, charts and graphs, videos, or audio lectures. In-class exercises and/or at-home assignments will allow students to practice content materials such as evaluating the .NET framework and platform and the social impacts of emerging technologies being developed in computer programming workspaces.

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

In both face-to-face and online modalities, students will use self-evaluation essays submitted through the accessible CMS to reflect on study skills and habits to be successful in technical education classes. Students will recognize their personal growth by analyzing



how computer programming concepts are used in applications in a variety of business scenarios. The course emphasizes thinking flexibly, critically, and considering different methods of solution by engaging in discussions, assessments, and writing activities.

Key: 1025