



CSIS-113B: JAVA PROGRAMMING - LEVEL 1

Effective Term

Fall 2023

BOT Approval Date

04/14/2022

Discipline

CSIS - Computer Sci/Info Systems

Course Number

113B

Course Title

Java Programming - Level 1

Short Title

Java Programming Level 1

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 COMP 122

Catalog Description

This course introduces the principles of object-oriented programming using the Java programming language. Students will investigate and evaluate various programming design methodologies and apply them to Java programming problems. Java features that will be covered include language syntax, class definitions, control structures, function 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. Analyze the basic structure of a Java application and be able to document, debug, compile, and run a simple application.
2. Create, name, and assign values to variables.
3. Demonstrate the use common control statements to implement flow control, looping, and exception handling.
4. Create methods (functions and subroutines) that can return values and take parameters.
5. Create, initialize, and use arrays to solve data storage and retrieval problems.
6. Explain the basic concepts and terminology of object-oriented programming.
7. Demonstrate the use of common objects and references types.

8. Create common GUI components and modify their properties at both run time and design time.
9. Differentiate between object-oriented, structured, and functional programming methodologies.
10. Analyze and trace the execution of Java computer programs.
11. Identify and assess the benefits of equity and inclusion in the tech workplace and society at large (e.g. teamwork, creativity, innovation, competitive advantage, etc.).

Learning Outcome Information

Course Learning Outcomes

Learning Outcome

Code and implement a looping construct for a given problem.

Develop algorithms that implement arrays for storage and data retrieval.

Demonstrate the use of functions for modular programming.

Code a complex (operations do not share the same level of precedence) algebraic expression in the target language.

Identify options and generate alternative approaches.

Engage in multiple and simultaneous outcomes and activities.

Program Learning Outcomes

Learning Outcome

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

Institutional Learning Outcomes

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.

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.

Content

Course Lecture Content

1. Java 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
 - b. Variables and method scope
 - c. Constructors
 - d. Overloaded constructors
6. Exception handling
 - a. Exception fundamentals
 - b. re-throwing exceptions
 - c. Custom exception classes
7. GUI Applications
 - a. Introduction to Swing
 - b. JLabels
 - c. JButtons
 - d. JTextFields
 - e. ActionEvents
 - f. MouseEvents
 - g. GridLayout
 - h. FlowLayout
8. Diversity, Equity, and Inclusion Issues related to Information Technology
 - a. Digital Divide
 - b. Racial and Gender Bias in Artificial Intelligence
 - c. The gender gap in programming and programmers

Methods of Instruction

Method

Discussion

Integration

Weekly discussion will be conducted as a means to discuss the digital divide, compare and contrast the use and implementation of the for, while, and do while looping constructs; the if, if else, conditional, and switch decision constructs; efficiency of algorithms that implement arrays for storage and data retrieval; debate the correct use of primitives to create abstract data types.

DE Adaptations for MOI

Weekly discussion conducted as a means to discuss the digital divide, compare and contrast the use and implementation of the for, while, and do while looping constructs; the if, if else, conditional, and switch decision constructs; efficiency of algorithms that implement arrays for storage and data retrieval; debate the correct use of abstract data types will be administered using the discussion tool supplied as part of the course management tool.

Method

Lecture

Integration

Lecture, with supporting visual materials (slide presentation or multimedia) will introduce conceptual and practical skills such as 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 the Racial and Gender Bias in Artificial Intelligence.

DE Adaptations for MOI

Lecture, with supporting visual materials will introduce conceptual and practical skills such as 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 the Racial and Gender Bias in Artificial Intelligence. Lectures will be delivered using video capture software to embed voice into presentation lecture slides. Internet technologies will be used to deliver additional lecture notes and sample materials. All lecture material will be delivered through the use of the course management system.

Method

Observation and Demonstration

Integration

Guided practice and demonstration will illustrate lecture principles and reading assignments with a focus on fundamental Java programming concepts and applying them to problem solving and potential solutions to homework assignments. 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; Discuss the gender gap in programming and programmers.

DE Adaptations for MOI

Guided practice and demonstration will illustrate lecture principles and reading assignments with a focus on fundamental Java programming concepts and applying them to problem solving and potential solutions to homework assignments. 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; discuss the gender gap in programming and programmers. Guided practice and demonstrations will be accomplished by using video/ audio capture software that will permit instructor demonstration to be captured and deployed. Hands-on activities can be similarly accommodated and supplemented with the use of software simulations. Guided practice videos will be delivered to the students through the course management system.

Methods of Evaluation**Method**

Exams/Tests

Integration

A midterm and final exam composed of short answer questions and small programming examples will be given. The exams will be designed to evaluate the mastery of such topics as looping constructs; decision structures; data storage and retrieval with arrays; and the creation of abstract data types. Exams 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.

DE Adaptations for MOE

A midterm and a final exam to evaluate the students understanding of 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 of primitives to construct complex types, and the development and proper use of abstract data type problems will submitted via the assignment editor of the course management system. Feedback to students along with evaluation will be given to the student via the course management system.

Integration

Discussion and guided practice consisting of Java 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 digital divide; the development and proper use of abstract data type problems will be required. Guided practice / discussion projects will be evaluated based on program completeness and how well directions are followed.

DE Adaptations for MOE

Discussion and guided practice consisting of Java 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 of primitives to create complex types; The digital divide; the development and proper use of abstract data type problems will submitted via the assignment editor of the course management system. Optional methods of delivering guided practice are through the use of the discussion, blog, and wiki tools of the course management system. Feedback to students along with evaluation will be given to the student via the course management system.

Method

Homework

Integration

Weekly programming assignments will apply important concepts and skills to the student's experience outside the classroom. The focus might be on 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 Artificial Intelligence and storage and data retrieval techniques using arrays. Weekly assignments 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.

DE Adaptations for MOE

In lieu of homework, students will complete assignments consisting of Java syntax and logic problems such as 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 of primitives to create complex types; racial and gender bias in Artificial Intelligence; the development and proper use of abstract data types. Assignments will be submitted using the assignment editor of the course management system. Optional methods for submitting assignments are through the use of the discussion, blog, and wiki tools of the course management system. Feedback to students along with evaluation will be given to the student via the course management system.

Assignments**Assignment Type**

Writing

In-class and/or outside of class

Outside of class

Formative or Summative

Formative

Assignment

Write a simple calculator program. 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. I want you to check to make sure you don't allow a divide by zero. Use an if else to do the checking. This check should only be done if the operation is division.

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'

How assignment will be adapted in online format

The assignment will be created using the assignment editor found in the course management system. The assignment will be distributed and collected using the assignments area of the course management system. Feedback will be provided to students via the course management system within five days of submission.

Assignment Type

Writing

In-class and/or outside of class

Outside of class

Formative or Summative

Summative

**Assignment**

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. Post your response in the appropriate discussion board. From two other student's posts write a response that add or expands on at least one fact about the figure you feel the author should have included. Please cite all sources with a link at the bottom of your posts and responses.

How assignment will be adapted in online format

This assignment will be administered through the assignments area of the course management system. Students will post their discussion responses to the appropriate discussion thread of the course management system.

Assignment Type

Reading

In-class and/or outside of class

Outside of class

Formative or Summative

Formative

Assignment

Read the chapter which discusses public and private inner classes and their usages. Once you have read the chapter go to the lecture material on class types. Read all the material and watch the videos that pertain to public and and private class types and how they are used. Once you have a good understanding of the material create a project that has a label that contains your name. Use three radio buttons that will allow the user to change the color of the text in the label to red, green, or blue. Use an inner class to handle the event for changing the color. In the comments of the code give an explanation of when inner classes should be used and why.

How assignment will be adapted in online format

This assignment will be administered and collected through the assignments area of the course management system. Lecture material will be delivered through text, graphics, and video and will be presented in pages that are part of the modules area of the course management system. Feedback for the assignment will be given in the grade book of the course management system.

Course Materials**Textbooks**

Deitel and Deitel, Java How to Program, Late Objects 11th Edition. January 2020

ISBN-13: 978-0134791401

ISBN-10: 0134791401

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.

Projects and assignments in Computer Science should include datasets and case studies that address diversity. Instruction should acknowledge the challenges that technology poses to diversity and equity (i.e. Digital Divide, racial and gender bias in Artificial Intelligence, etc.), as well as the opportunities that technology affords in confronting and overcoming racial, gender, and other bias. Diversity and inclusion is infused in the course through an acknowledgement of biases and prejudices that may exist, and a holistic view that each individual can make a positive impact in combating bias and prejudice in the workplace and society in their sphere of influence.

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

Instruction will include diverse view points and should include readings and videos by authors that promote diversity and inclusion, and should highlight achievements of diverse individuals in the Computer Science field, both historical, and present day.

Project 2 is a group research / writing project. This activity takes into account the context of the student in terms of interests, identities, cultural and linguistic practices and histories. This project also welcomes the identities, backgrounds, perspectives of the diverse student population without barriers or judgement.

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

Each lecture of the course will have several small programs to solve. Students are encouraged to work in groups to solve the problem as a team. In addition, two group projects will be assigned where students will be required to work with each other to derive a solution. Each unit of the course has its own discussion board where students can go to interact and help each other understand topics such as the differences between complex and primitive data types, branching, looping, pointers, arrays, public and private interfaces.

To further promote student to student interactions a discussion board will be created in the course management system titled "Student Lounge" where students can go to converse and share ideas with each other.

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

A welcome letter will be sent to the students the first day of class which not only welcomes students to the class but also gives details of how to navigate the course. The announcements area of the content management system will be used to send weekly announcement about important concepts and due dates. A welcome to class announcement will be provided which contains a video that explains the details of the class.

A separate discussion board will be created in the course management system for each unit of the class. These discussion boards are a place where students can get feedback from the instructor on such topics as the difference between primitive and complex data types, arrays, pointers, functions, branching, and repetition.

In addition, a general questions discussion board will be provided where students are encouraged to speak with the instructor on any programming related topic.

Students are encouraged to email the instructor with any questions they may have that they do not want publicly published.

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

All courses have an online lecture component. This is in lieu of purchasing a textbook. The content of the course has several "test your knowledge" pieces where students get instant feedback about their understanding of the material presented. In addition, a series of "video shorts" are given that reinforce topics given. Practice exams will be created as a way for students to get an understanding of the type of questions be asked on the exams and a way to practice skills before taking the exam. At the end of each lecture component several practice problems will be presented to give students a better understanding of the material offered in that unit. All practice problems will have the solution with the so students can reflect on their coding techniques and logic used to solve the problem.

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

The course emphasizes the idea of thinking logically and to consider other logical interpretations of problem solving using a programming language. Through discussion and assignment, we ask students to pose questions about problem solving and how they approach problem solving through the application of a programming language. Through self-assessment we ask students to reflect on where they are successful and where they are having difficulty and how they can improve upon both their syntactical and logical skills.

We encourage students to think outside the box and investigate other problem solving techniques from outside sources as a means to not only improve but to understand problem solving from other points of view.