

MONTGOMERY COLLEGE: Course Syllabus

Computer and Data Sciences Department CMSC100: Introduction to Programming: 202724-50039, Prof. J. Joy

Fall, 2026 Starts Monday, December 31, 2026; ends Friday, January 22, 2027

This class does NOT have a meeting time. It is completely asynchronous.

I. Contact Information: Professor J. Joy

Email: Janet.Joy@montgomerycollege.edu (*This is the preferred way to contact me.*) I usually answer my email first thing in the morning and again in the evening. (*It depends on my schedule.*) On weekends it may be less often.

Questions: If you have a personal question please use the email above or course mail. If you have a general question that is of interest to the whole class, such as clarification of an assignment, please use Questions in the discussion area of Blackboard.

Office Hours: Online with Zoom: <https://zoom.us/j/4497759354>

Office Hours for Spring and Fall: Monday 11am to 12 noon, Thursday 3pm to 5pm. No appointment necessary for these office hours.

Please be aware that you are sharing these office hours with other students and classes.

When you type in your name, please type your course and section after so that I know which class you belong to.

Office Hours during MC Holidays, Winter, and Summer sessions are by appointment only.

By appointment only: If you need to see me, please send me 2 or 3 blocks of time you could meet and I will get back to you with a time that works for both of us.

In addition, I occasionally have optional class "Zoom coffee breaks" that are at varying times each week. These are entirely optional and informal opportunities to say hello to me and some of your classmates or to ask questions.

Your Montgomery College e-mail account is the official means of communication for the college. **I may not receive emails sent from outside addresses.** Blackboard will use this email address to send reminders about overdue projects and other announcements. It is recommended that you check this account routinely. To check your e-mail, log into your MyMC Online account and locate the e-mail icon in the upper right hand corner of the page. You can forward your MC email to your other email. Announcements sent from Blackboard may have "Do not respond" as the subject. Please take a look to see if it is important!

II. General Course Information

CMSC 100 - Fundamentals of Programming

Designed for students with no prior programming experience, this course introduces students to fundamental structures of sequence, selection, and repetition, emphasizes solving simple problems using a flowchart. With a high-level language, students code, test, and debug short programs. *Assessment Level(s): ENGL 101/ENGL 101A, MATH 050, READ 120. Two hours each week. Formerly CS 100.2 semester hours.*

Accessibility: This section uses Alice3 to create 3D animation. This course is not recommended for students who cannot view and create videos. The videos used in this course have closed captions or a text version is available. Montgomery College accessibility statement: <http://cms.montgomerycollege.edu/edu/Department.aspx?id=53990>

Preparedness: This is a fully online class. To succeed in this course you should be confident working with a computer, accessing information via the Internet, and using email as a primary means of communication. You should be comfortable with email attachments, troubleshooting an Internet connection, and downloading software. Online courses require extreme self-discipline. One must log on 3-5 times per week and be prepared to read and follow through on assignments and instructions. Students must plan to spend 4-6 hours per week preparing and submitting assignments. Initially, a great deal of time is spent becoming familiar with Blackboard and dealing with technical problems. Technology is unreliable. The plan to submit homework at the last moment can be defeated with a busy or down server.

III. Student Learning Outcomes

Upon course completion you will be able to:

By the end of this course you will be able to:	You will demonstrate this objective through:
1. Define the software development life cycle.	By creating a storyboard for your final project and testing and debugging each of the programming assignments.
2. Describe the core features of programming languages: a) Variables, constants, and data types b) Flow control structures c) Arrays	Each week you will learn new features. You will test your understanding through quizzes. Then you will apply what you have learned by writing programs that use these core features and the ones you have learned previously.
3. Demonstrate the ability to design, develop, and test basic computer programs using the core programming features.	This is demonstrated in all programming projects, progressing from simple to more complex. In your final project you will apply these skills to create an animated movie with sound and other effects.
4. Decompose a software program into functional subprograms (procedures and functions).	As your programs get larger, you will use functions and procedures in order to work on one part at a time. This will make it easier to test and debug your programs.

Learning to program is a cumulative skill. Each week you will learn a few new skills or tools. You will share your thoughts, and what you have created using these new skills in the discussion area in Blackboard. You will also have a chance to see what your classmates have created and get new ideas and techniques from them. You will test your understanding of the new material through quizzes. Finally, you will **apply** what you

have learned by creating a program that uses the new tools, along with the skills from previous weeks to create increasingly more complex 3D animation. You will use a variety of control structures to make interesting animations. The control structures allow you to specify which actions to take, whether to perform actions at the same time or in sequence and to specify how many times an action should be performed. As your programs get larger, you will learn to break them into subparts so that you can write, test and debug one piece at a time.

IV. Required materials

There is NO required text for this class. We will use online materials and tutorials that are available for **free** on the internet. Alice is available for both the Mac and Windows computers.

- Visit [Alice.org](http://www.alice.org) and read all of the installation information for your computer.
- Download and install Alice 3.9, or latest version from <http://www.alice.org> Do NOT download Alice 2.x

Access to Alice software is required! If you cannot install Alice on your computer, you may use Alice in any of the MC computer labs. This is not an excuse for not doing the work.

Alice 3.x © 2008-2016, Carnegie Mellon University. All rights reserved.

Alice/Zebra0.com © 2016, Joy, J. for zebra0.com. All rights reserved.

V. Grading

Grades are earned, not given. Students may be asked to explain their work orally on Zoom.

Course grades will be based upon the following:

- There is a final project for this course in lieu of exams. You start the final project on the first day, then add to it after every lesson. You will be required to complete a summary of your final project and evaluate 3 other projects by your class peers.
- There will be 10 programming assignments. All assignments must be uploaded to your course assignment page to receive credit.
- All assignments, discussions and quizzes will be due at the end of the unit, postmarked by midnight.
- I usually grade assignments within 2 days once submitted. If I ask for corrections, they must be resubmitted within 1 week of the due date. The last project may not be resubmitted. At that point in the semester, you should be used to using the checklist.
- You will not be able to submit assignments or corrections one week after the due date. Any assignments not submitted within 1 week of the due date will receive 0.
- Do not email the assignment for "pre-grading". Submit it for a grade, then correct any errors and resubmit within 1 week of the due date.
- Each unit has discussion questions in Blackboard. You will be required to give an initial response during the first half of the unit (usually Wednesday) and you must respond to at least two other students regarding their initial postings. Responses must be substantive, relevant and constructive. Responses must be at least 25 words long using standard academic English. If there are spelling or grammar errors in the initial post you will receive a 0. I drop the 3 lowest discussion grades, therefore I do not reopen discussions if you miss one.

- I grade the initial post, then I grade again after the discussion is closed to check for interactions with classmates.
- There will be a quiz in Blackboard every 2 or 3 days. I drop the 2 lowest quiz grades, therefore I do not reopen quizzes if you miss one.

Assignment	Each	Total %
10 Programming projects	5%	50%
12 Discussion questions	1%	11%
12 Quizzes	1%	11%
Final project design		8%
Final Project	required	20%

A=100-90% B= 89-80% C=79-70% D=69-60% F=60%-below

Please note that Blackboard has a column “Unweighted total” Disregard this column!

The grade is calculated as shown above.

The column labeled “Overall” shows your grade so far. At the end of the semester, this is your final grade.

This is an Online course. Active participation in the online activities and completion of all homework and Online assignments is required in order to pass this course.

Due Dates: All due dates are in Blackboard. You can check the calendar in Blackboard.

Participation: Students must participate in the Blackboard discussion about 3 or 4 times per week. **If you miss 2 discussions in a row without contacting me, you are subject to being dropped from the class.**

Audit Policy: If you are auditing, you are welcome to participate in the Blackboard discussions and take all quizzes and submit programs but it is not required.

VI. Class Policies

Important Student Information Link

In addition to course requirements and objectives that are in this syllabus, Montgomery College has information on its web site (see link below) to assist you in having a successful experience both inside and outside of the classroom. It is important that you read and understand this information. The link below provides information and other resources to areas that pertain to the following: student behavior (student code of conduct), student e-mail, the tobacco free policy, withdraw and refund dates, disability support services, veteran services, how to access information on delayed openings and closings, how to register for the Montgomery College alert System, and finally, how closings and delays can impact your classes. If you have any questions please bring them to your professor. As rules and regulations change they will be updated and you will be able to access them through the link. If any student would like a written copy of these policies and procedures, the professor would be happy to provide them. By registering for this class and staying in this class, you are indicating that you acknowledge and accept these policies.

<http://cms.montgomerycollege.edu/mcsyllabus/>, [Additional Montgomery College policies](#)

VII. Support

Computer problems: As a computer student, you are expected to anticipate potential computer problems. Save often! Keep backups! Allow plenty of time to complete the assignment! Computer problems are not an excuse for submitting an assignment late! I can provide help if you send me a clear explanation of the problem, plus any relevant source files or screen shots.

Netiquette: Etiquette rules for the discussion board.

The discussions are an important part of online classes. Each week you will share ideas with your classmates. You can learn a lot from your classmates and by sharing ideas. Your classmates come from many different cultures and backgrounds. You want to share ideas and tips, not offend. Please read <http://www.zebra0.com/MC/netiquette.pdf> for discussion rules.

Technical Requirements & Technical Support: You will need the following to participate online:

- Regular use of a computer with Internet access and a web browser such as Firefox, Chrome, or Internet Explorer. Expect to spend several hours online each week.
- A web browser such as Firefox, Chrome, or Internet Explorer.
- Speakers or headphone to listen to the videos and to play the sounds you add in Alice.
- See prepare yourself: <http://cms.montgomerycollege.edu/distance/prepare/>
- It is highly recommend that you have internet access at home, however, there are computer labs <http://cms.montgomerycollege.edu/oit/InTech.aspx?id=60795>

For technical assistance with college supported resources, call the Montgomery College IT Service Desk at 240-567-7222 or [://cms.montgomerycollege.edu/EDU/Department2.aspx?id=9356](http://cms.montgomerycollege.edu/EDU/Department2.aspx?id=9356)

Blackboard Help Desk: The **HELP** link on the left-hand course menu links to the **MC Blackboard Online Support Center**:

- Call the Support Center at 240-567-7222 or
- Chat with a service representative, or
- Submit a ticket.

Note: Click the **My Support** link at the top of the Blackboard Online Support Center screen to view a history of your correspondence with the Blackboard Support Center.

System Downtime: The Office of Information Technology conducts computer network maintenance on Sunday morning from 12:01 AM to 6:00 AM each week. During this time you may be not be able to access My MC to login to Blackboard. Do not rely on this time to submit course work.

Distance Learning Support: For all general distance education related questions, contact the Office of Distance Education and Learning Technologies at 240-567-6000 or dl@montgomerycollege.edu. For all Blackboard and MyMC related questions and issues, contact the IT Service Desk at 240-567-7222 or ITServiceDesk@montgomerycollege.edu or [Blackboard Online Support Center](#).

VIII. Course Resources & Technologies

Adobe Acrobat Reader, required: Download and install from <https://get.adobe.com/reader/> Copyright © 2017 Adobe Systems Incorporated. *You probably already have this if you are reading the syllabus. No privacy policy available. Adobe acrobat is accessible with a screen reader.*

Alice 3.1, required: Download and install Alice 3.2 or later from <http://www.alice.org>, Alice 3.x © 2008-2017, Carnegie Mellon University. You do not need to login or register. Alice is available for PC and Mac. No privacy policy is available. Alice is a tool for creating 3D animation. There is no comparable tool for students who cannot see the animation.

Alice Lessons at Zebra0.com, required: Available at <http://zebra0.com/alice>

Author: Janet E. Joy; Publisher: Zebra0.com, This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#)

Privacy policy: <http://www.zebra0.com/resources/privacy.php> All videos have closed captions.

Blackboard, required: Copyright © 1997 - 2017. Blackboard Inc. Login to Blackboard from MyMC. Privacy policy: <http://www.Blackboard.com/footer/privacy-policy.aspx> Blackboard is fully accessible.

Eclipse, optional: Download and install from <https://eclipse.org/downloads/> Copyright © 2017 The Eclipse Foundation. *If you have a problem installing Alice with the message that you need the Java JDK, install Eclipse and then retry installing Alice.* Eclipse is accessible. Privacy police: <https://eclipse.org/legal/privacy.php>

Screencast-O-Matic, recommended: Download and install or run online from <https://screencastomatic.com/> © Screencast-O-Matic , privacy policy: <https://screencast-o-matic.com/privacy> You may use any video recording software except Flash to create the required videos for this class. Screencast-O-Matic is a tool for creating videos. It is vision dependent.

Zoom, required: Copyright ©2017 Zoom Video Communications, Inc. Join from PC, Mac, iOS or Android: <https://zoom.us/j/4497759354> *No privacy policy available. Eclipse is accessible.*

IX. A Typical Week

An Online class requires quite a bit of self-motivation. All of the projects and activities are due on Sunday night, so that we end one week on Sunday and begin the next week on Monday.

When you make out your schedule for the week, be sure to block out enough time each week when you can read, study, work on a computer! In a class that meets fewer than 15 weeks, there will be several assignments, quizzes and discussions due each week. Although they are mostly due on Sunday night, you should try to spread them out over the course of the week, depending on your own work schedule. Your initial posts in the discussion, and first attempt at the quizzes are due on Wednesday.

Here is the math: Being a full-time student (*15 credits or 5 classes*) is considered a **full time job**. A full-time work week is 40 hours. Thus, one college class will requires about 8 hours per week in a 15 week semester, or 120 hours for the semester. A shorter term requires the same amount of time, assignments and effort. Use the table below to block out sufficient time for each course:

Weeks in Class	Hours per week	Hours per course
15	8	120
10	12	120
8	15	120
5	24	120
4	30	120

Start on Monday by looking in Blackboard for the week's assignments and discussion questions. Keep the discussion questions in mind as you read the chapter. Some weeks cover more than one unit or chapter.

Write your initial post to the discussion questions by Wednesday. Read the other posts and exchange ideas and thoughts with classmates during the rest of the week.

Each unit has a quiz in Blackboard, The first attempt is due on Wednesday. You have 2 more attempts until Sunday. Take the quiz again if you miss any questions.

Check into the discussion board in Blackboard periodically to ask questions, answer questions, and respond to your classmates.

You are expected to save all of your work on a USB drive or other storage device. You are responsible for completing all of the work on time even if your computer crashes.

Class Schedule and Important Dates

In order to provide the best possible learning experiences, these dates may change. Please refer to the resources in Blackboard for any announcements or changes. All of the learning modules are required.

Course is from Monday, December 21, 2026 to Friday, January 22, 2027

Week 1: Monday, 12/21/26 to Sunday, 12/27/26

Study:

**CMSC
100**

[Introduction to CMSC100](#)

Here, you will learn all of the skills necessary to succeed in this course.



[Introduction to Alice 3](#)



[Your First program: Hello World](#)

In this lesson, you will learn the basics of Alice so that you can set up a scene and explore the characters and other objects that will make up your virtual worlds. You will also create a program with a 3D character who says "Hello World"



[The Software-Lifecycle](#)

In this lesson you will learn to design large programs using top down design, and various other tools. You will break the program into small manageable parts to facilitate writing, testing and debugging the program. In order to do this you will learn to use several tools traditionally used by programmers. You will use all of these tools to design your final project.



Due Wednesday, 12/23/26

Initial post in discussion 1: Introductions
 Initial post in discussion 2: Reaction to Alice
 Initial post in discussion 3: Motivation
 First attempt at quiz 1: Syllabus
 First attempt at quiz 2: Introduction to Alice
 First attempt at quiz 3: The Software-Lifecycle



Due Sunday, 12/27/26

[Final Project : Step 1: Proposal](#)

[Final Project : Step 2: Algorithm, Storyboard, and Sound](#)

[In Blackboard: Check major and submit Academic Integrity statement](#)

[Program 1: Hello World](#)

Week 2: Monday, 12/28/26 to Sunday, 1/3/27

Study:



[Start, Save, Reopen and modify your program](#)

In this lesson, you will learn to modify an existing program and use some of the templates that give you a starting scene.



[Understanding the Alice Virtual World](#)

In this lesson, you will gain a deeper understanding of objects in the Alice virtual world.



[Setting up the Alice Virtual World](#)

In this lesson, you will fine tune your ability to position objects in the Alice virtual world.

There are a number of tools in the setup scene window that allow you to make changes to the virtual world. We will discuss some of these tools.



[Variables in Alice 3](#)

A variable is a named location in memory. In this lesson, you will learn about variables and how to use them. Your program will declare variables, assign them values, then use them in your code.



Due Wednesday, 12/30/26

Initial post in discussion 4: Storyboard
 Initial post in discussion 5: Hello World
 Initial post in discussion 6: Variables
 First attempt at quiz 4: The Alice Virtual World



Due Sunday, 1/3/27

[Final Project : Step 3: Locations](#)

Week 3: Monday, 1/4/27 to Sunday, 1/10/27**Study:****Posing Objects in Alice3**

So far, we have moved the objects around using move, turn and roll. If we want them to walk, kick, or put their hands on their hips and say "How dare you!" we need to change the position of the subparts (elbow, hip, etc.) An objects subparts can be positioned using turn and roll. *(You cannot move a subpart as that would separate it from its body.)*

Each subpart has its own orientation and sense of direction. The sense of direction for subparts may not be the same as its parent object.

Subparts can be positioned at design time in the scene editor or during execution using code.

In this lesson, you will learn to move and pose the Alice 3D characters and the subparts such as arms and legs. You will do this in both the scene setup mode and using code.

**Scene Procedures in Alice 3**

A movie may have many scenes such as introduction, ending, and everything in between.

In the scene editor, you may create different scenes by arranging one scene on the left and one on the right, perhaps another in the middle.

In your code you will create a scene procedure with all the comments and code that belongs in the scene.

In myFirstMethod, you will call each of the scenes.

**User Input in Alice3**

In this lesson, you will learn to create an interactive program.

Your program will ask the user his name and then make a 3D character say "hello" using the name they entered.

**A Little Math**

In this lesson, you will use arithmetic operators to do some simple math.

**Due Wednesday, 1/6/27**

Initial post in discussion 7: Poses

First attempt at quiz 5: Poses in Alice3

First attempt at quiz 6: Hello World & Scene Procedures

First attempt at quiz 7: Variables

**Due Sunday, 1/10/27**

Final Project : Step 4: Scene Procedures

Final Project : Step 5: Interactions

Program 2: A Comic

Program 3: Talk to the User

Program 4: Math

Week 4: Monday, 1/11/27 to Sunday, 1/17/27**Study:****TRUE OR****FALSE?** **Boolean Expressions: true or false**

Sometimes a programmer would like one statement, or group of statements to execute only if certain conditions are true. There may be a different statement, or group of statements that are to be executed when the condition is false. An expression that can be evaluated to true or false is called a Boolean expression. (Named after a mathematician named Boole who developed Boolean algebra.)

In this lesson, you will use Boolean Expressions. Boolean expressions are ones that can be evaluated as true or false.

You will use if/else control structures to control which code to execute.

You will perform tasks such as asking the user a question and tell if the answer is right.



Creating Animation in Alice3

Animation is created in Alice by moving objects a little bit at a time using several different control structures.



Class Procedures in Alice3: Reusable Code

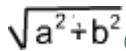
In this lesson, you will create procedures. Procedures let you write code for an object to perform an action.

For instance, you can write a procedure for a biped to wave. Then, whenever you want a character to wave, you just call the procedure.

The code can then be used many times and for many objects. This will allow you to decompose a program into functional subprograms and create and test larger programs.

We will also use parameters so that we can tell the procedure how fast to wave, or how many times to wave. We can also use parameters to tell who the bunny faces when they wave.

This is a very powerful tool and will let you greatly improve the efficiency of your code.



Calculations and Functions

In this lesson, you will use arithmetic operators to move the characters to precise locations, to find totals, and other calculations.



Due Wednesday, 1/13/27

Initial post in discussion 8: Boolean Operations

Initial post in discussion 9: Animation

Initial post in discussion 10: Procedures and Random Numbers

First attempt at quiz 8: Arithmetic and Boolean Operations

First attempt at quiz 9: Animation

First attempt at quiz 10: Class Procedures



Due Sunday, 1/17/27

Final Project : Step 6: Biped Procedure

Program 5: A Quiz Game

Program 6: Animation

Program 7: An Exercise Class

Week 5: Monday, 1/18/27 to Friday, 1/22/27

Study:

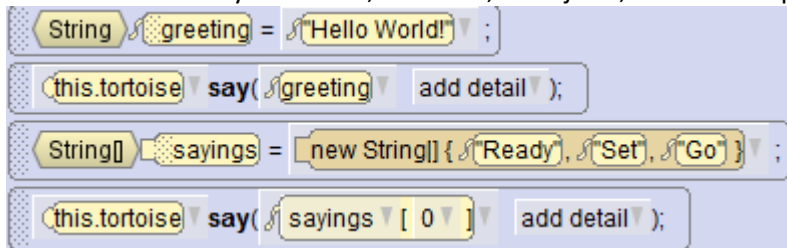


Arrays: Working with lists of objects

An array is a list. When you declare a variable in Alice, you can select to make it an array.

In this lesson, you will declare and use arrays. An array is a variable that is a list.

You will create arrays of words, numbers, or objects, then use loops to process the entire array.

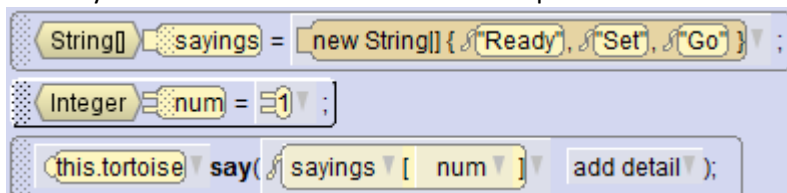


The variable `greeting` is not an array, it is a scalar. It has a single value, "Hello World". When the tortoise says `greeting`, he says "Hello World"

The variable `sayings` is an array. The items in an array are numbered starting with 0. The number is called the index or subscript. When you use an array you need to use a subscript to specify which item, or element, in the list is to be used.

In this example, the tortoise says `sayings[0]`, of "Ready".

An array can also use variables as the subscript:



In this example, `num` is 1, so when the tortoise says `sayings[num]` he says "Set".

In this module, you will declare and use arrays. An array is a variable that is a list.

You will create arrays of words, numbers, or objects, then use loops to process the entire array.



Random Numbers

In this lesson you will learn more about random variables. We can use random numbers in many ways. One thing we can do with random users is to flip a coin or select a card from a deck. Those are things that we think of as random.

We can also use random numbers to make movement seem more natural. When a person jumps up and down, they don't jump exactly the same amount each time, so we could have them jump a random amount within a range.



Events: Responding to the Mouse and Keyboards

Events are things that happen. In programming, some of the events that can occur are clicking or moving the mouse, typing a key on the keyboard, or other user interaction.

When these events occur, nothing will happen unless you tell the program to listen for these events and what to do when these events occur.

If we want to execute some statements when one of these events occurs, we have to tell Alice to "listen" for that event. We do this by adding a listener. We tell it what event to listen for, and what function to execute when that event occurs.

After adding an event listener, that event will trigger the execution of a procedure or function. Programmers refer to this as "Event Handling."

In this module you will learn to use event listeners to respond to the mouse and keyboard, add drag and drop and other interactive features.



Final Project Presentations



Due Wednesday, 1/20/27

Initial post in discussion 11: Arrays

Initial post in discussion 12: Present your final project in Blackboard

First attempt at quiz 11: Calculations and Random Numbers

First attempt at quiz 12: Arrays, Advanced Techniques, and Events



Due Friday, 1/22/27

[Final Project : Step 7: Arrays](#)

[Final Project : Step 8: Events and Alternate Ending](#)

[Program 9: Arrays](#)

[Program 8: Jumping](#)

[Program 10: Events](#)

[Final Project](#)

Updated October 23, 2026