

**While from a previous semester, this syllabus is indicative of the course content/assignments.**

# CS 170/1-6, Introduction to Computer Science I

## Syllabus

Jeff Epstein

Fall 2023

## Contents

<b>1</b>	<b>Basic information</b>	<b>2</b>
1.1	Course description . . . . .	2
1.2	Learning outcomes . . . . .	2
1.3	Logistics . . . . .	2
1.4	Contact . . . . .	2
1.5	Prerequisites . . . . .	3
1.6	Textbook . . . . .	3
1.7	Schedule . . . . .	3
<b>2</b>	<b>Coursework</b>	<b>4</b>
2.1	Grading rubric . . . . .	4
2.2	Homework . . . . .	5
2.3	Quizzes . . . . .	5
<b>3</b>	<b>Technology and tools</b>	<b>5</b>
3.1	Java Development Kit . . . . .	6
3.2	Text editor . . . . .	6
3.3	Canvas . . . . .	7
3.4	JQuiz . . . . .	7
<b>4</b>	<b>Policies</b>	<b>7</b>
4.1	Classroom expectations . . . . .	7
4.2	Communication . . . . .	8
4.3	Deadlines, absences, make-ups, and late submissions . . . . .	8
4.4	Regrades . . . . .	8
4.5	Getting help . . . . .	9
4.6	Academic integrity . . . . .	9
4.7	Inclusivity statement . . . . .	13
4.8	Academic accommodations . . . . .	13

# 1 Basic information

## 1.1 Course description

An introduction to Computer Science for students expecting to utilize serious computing in course-work, research, or employment. Emphasis is on computing concepts, programming principles, algorithm development and basic data structures, using the Java programming language and Unix operating system.

## 1.2 Learning outcomes

Students completing this course successfully will be able to (1) design simple computer algorithms; (2) program these using the Java language; (3) understand the core elements of imperative programming; (4) execute, test, and document programs; (5) know when and how to use simple variables, one-dimensional and two-dimensional arrays; (6) understand and program simple recursive algorithms; and (7) know how to detect errors in programs, use debugging tools, and effectively correct bugs.

## 1.3 Logistics

Meeting times are as follow:

Section	Time	Place	Instructors
Lecture	MoWe 1pm-2:15pm	M&SC E208	Epstein
Lab 1	Fr 8am-8:50am	M&SC E308A	Yao, Odonga
Lab 2	Fr 9am-9:50am	M&SC E308A	Yao, Odonga
Lab 3	Fr 1pm-1:50pm	M&SC E308A	Hairston, Campbell
Lab 4	Fr 11am-11:50am	M&SC E308A	Hairston, Li
Lab 5	Fr 3pm-3:50pm	M&SC E308A	Kobara, Campbell
Lab 6	Fr 4pm-4:50pm	M&SC E308A	Kobara, Li

Please attend only the section for which you are registered.

## 1.4 Contact

The principal instructor for this course is:

Name	email	Office hours
Jeff Epstein	jeff.epstein@emory.edu	Mo 4pm-5pm, M&SC E421

The TAs for this course are:

Name	email	Office hours	Office hours location
Seibi Kobara	seibi.kobara@emory.edu	Th 1pm-2pm	Zoom
Kiersten Campbell	kiersten.campbell@emory.edu	Tu 3pm-4pm	Zoom

Name	email	Office hours	Office hours location
JaMor Hairston	jamor.hairston@emory.edu	Th 4pm-5pm	Zoom
Timothy Odonga	timothy.odonga@emory.edu	Fr 2pm-3pm	Zoom
Siyu Li	siyu.li@emory.edu	Mo 4pm-5pm	M&SC E308
Danwei Yao	danwei.yao@emory.edu	We 4pm-5pm	Zoom

Office hours begin the second week of class.

When contacting a member of the teaching staff by email, please use the course code (“CS 170”) in the subject, so that we know the context.

Please do not send source code as an email attachment. Assignments are not accepted by email.

## 1.5 Prerequisites

No previous computer science knowledge is expected or necessary.

## 1.6 Textbook

There is no required textbook for this course.

However, we do recommend the book “Think Java @ Emory” by Valerie Henderson Summet. You can find this book linked from our course site.

## 1.7 Schedule

We present a tentative course schedule. Topics and dates are subject to change without notice.

Week	Date	Note	Topic
1	23 Aug 2023		syllabus, compiling
	25 Aug 2023		NO LAB
2	28 Aug 2023		directory hierarchy, squares with turtle, for loops
	30 Aug 2023		more nested for loops, methods, parameters
	1 Sep 2023		NO LAB
3	4 Sep 2023	LABOR DAY	
	6 Sep 2023		functional decomposition (House example), Java values
	8 Sep 2023		quiz00
4	11 Sep 2023		more values, Math, commenting, indentation, parameters, variables
	13 Sep 2023		for loops really (arbitrary-sided polygons), increment operators, accumulators
	15 Sep 2023		quiz01
5	18 Sep 2023		more nested loops, intro to booleans and conditions
	20 Sep 2023		more booleans, methods with return values, accumulators
	22 Sep 2023		quiz02
6	25 Sep 2023		if else
	27 Sep 2023		if else review with FizzBuzz, string manipulation, char
	29 Sep 2023		quiz03

Week	Date	Note	Topic
7	2 Oct 2023		string practice, binary
	4 Oct 2023		binary practice, while, intro to arrays
	6 Oct 2023		NO LAB
8	9 Oct 2023	FALL BREAK	
	11 Oct 2023		arrays
	13 Oct 2023		quiz04
9	16 Oct 2023		arrays, value vs reference parameters
	18 Oct 2023		parallel arrays, command-line arguments, int_max, filter pattern, substring s
	20 Oct 2023		quiz05
10	23 Oct 2023		image manipulation
	25 Oct 2023		2d arrays
	27 Oct 2023		quiz06
11	30 Oct 2023		triangular matrices, ragged 2d arrays, more images (blur, scale)
	1 Nov 2023		recursion, fractals
	3 Nov 2023		quiz07
12	6 Nov 2023		fractals, random numbers
	8 Nov 2023		
	10 Nov 2023		quiz08
13	13 Nov 2023		
	15 Nov 2023		file IO
	17 Nov 2023		quiz09
14	20 Nov 2023	TURKEY DAY	there IS A lecture here
	22 Nov 2023		
	24 Nov 2023		NO LAB
15	27 Nov 2023		exceptions
	29 Nov 2023		exceptions
	1 Nov 2023		quiz10
16	4 Dec 2023		debugging

## 2 Coursework

### 2.1 Grading rubric

Your final numeric grade will be calculated with the following composition:

Component	Weight
Quizzes	70%
Homework	30%

There is no final exam in this course.

Your final numeric grade, expressed as a percent of the maximum possible numeric grade, will be converted to a final letter grade based on the following table:

<b>Letter</b>	A	A-	B+	B	B-	C+	C	C-	D+	D	F
<b>Minimum</b>	93.3	90	86.6	83.3	80	76.6	73.3	70	66.6	60	0

## 2.2 Homework

Homework is intended to let you practice your new skills and knowledge and to prepare you for the quizzes. You'll have at least a week to work on each homework. You can complete each homework on your own computer or on a lab computer. Homework can be accessed on the JQuiz system.

Each homework consists of a number of different questions of several different types: code reading, code writing, code debugging, and free response.

Each homework has the same weight: your overall homework grade is the average of all your homework grades. Your lowest homework grade will be dropped.

## 2.3 Quizzes

Quizzes represent an opportunity to prove your new skills and knowledge, after having already practiced with homework. The time allowed for a quiz is limited to one lab session. Quizzes will be administered in a proctored environment during lab sessions, and only on the university's lab computers. During the quiz, use of outside resources, use of personal computers or phones, communication between students, and use of any other program or web site is prohibited. Quizzes can be accessed on the JQuiz system.

Each quiz consists of a number of different questions: multiple choice, code reading, code writing, code debugging, and free response.

Each quiz has the same weight: your overall quiz grade is the average of all your quiz grades. Your lowest two quiz grades will be dropped.

In order to get credit for a quiz, you must be physically present in the lab during the administration of the quiz, and must present your Emory ID card to a proctor upon request. Failure to comply will result in receiving a grade of zero for that quiz. Attempting to circumvent these policies may constitute a violation of academic integrity.

Quizzes will be held in the computing lab. While in the lab, students are obliged to comply with the lab rules.

## 3 Technology and tools

Each student should have consistent access to a personal desktop or laptop computer with either Linux, Mac OS, or Windows operating system, and a stable internet connection.

Please consult Emory's recommendations on setting up your computer and getting access to Emory services.

Students are responsible for maintaining their computing equipment in good working condition at all times. Technical issues with personal computer equipment and internet connection are not valid reasons for requesting deadline extensions or other accommodations.

In addition, we will use the following tools:

### 3.1 Java Development Kit

We will use the Java Development Kit (JDK) for compiling and running programs. You will need to have this software installed on your computer in order to complete assignments. Please follow these instructions to download and install the JDK:

- 1) Go to the download page.
- 2) Select the tab matching your computer's operating system: Linux, Mac OS, or Windows.
- 3) Click the link to download and install the kit:
  - For Windows, choose the “x64 installer” or “x64 MSI installer.”
  - For Mac OS, choose the “DMG installer”, in either ARM64 (for newer Apple Silicon-based Macs) or x64 (for older Intel-based Macs).
  - For Linux, choose the package that matches your platform.
- 4) Launch the installer and follow the instructions.

To use the JDK, you'll need to open a terminal on your computer:

- On Windows, type `cmd` into the search bar and press enter. This should open a Command Prompt window.
- On Mac OS, run the Terminal program in your Applications/Utilities folder.
- On Linux, open a terminal as required by your distribution.

Once in a terminal, you can type the commands `javac` and `java` to run the Java compiler and interpreter, respectively. You can use the `exit` command to close the terminal.

### 3.2 Text editor

You will need a text editor for writing and editing code. Please note that a text editor is not a word processor: a tool such as Microsoft Word, Google Docs, or Apple Pages is not acceptable. A text editor must be able to save documents in plain text format. Many text editors are specifically oriented towards programmers and provide additional functionality, such as syntax highlighting. Here are some suggestions:

- Sublime Text – Excellent text editor for programmers. Downloadable for Linux, Mac OS, and Windows.
- Lapce – Another great cross-platform text editor.
- Gedit – Comes with many Linux distributions and can also be downloaded for Mac OS and Windows.
- Notepad++ – Popular text editor only for Windows.
- Atom – Now defunct editor for most platforms.
- TextMate – Downloadable editor only for Mac OS.

Alternatively, you can use an Integrated Development Environment (IDE), such as VS Code or Eclipse. However, these options are not supported by this course's teaching staff.

### **3.3 Canvas**

We will use Canvas for discussion and course information. For questions about Canvas, please consult Emory's official Canvas support service.

Students are encouraged to make use of Canvas's discussion forum to ask and answer questions about course material. The principal instructor, the TAs, and other students will do their best to help you.

A one-point final grade bonus is offered to the five students with the highest number of helpful answers on the Canvas discussion forum.

Although you are free to use the discussion system to discuss course material, posting homework or quiz solutions is prohibited by the course's academic integrity policy. Posting inappropriate comments on the discussion forum may result in a grade penalty.

### **3.4 JQuiz**

Homework and quizzes will be administered on JQuiz. You should receive an email with your login credentials. If you can't log in, please contact the course instructor. You can access JQuiz on a lab computer, as well as on your personal computer when it is connected to the Emory wireless network; in order to access JQuiz from home, you must first log in to the Emory VPN.

Homework and quiz assignments will be posted on JQuiz, under the "Assignments" tab. You will be able to submit homework solutions on JQuiz until the due date. Quizzes are accessible on JQuiz only from the computer lab and only during the time period of the lab session.

JQuiz allows you to run your code directly in your web browser. This makes it convenient to test your homework and quiz solutions prior to submission. However, be aware that running code in JQuiz will necessarily be slower than running your code with the JDK. Therefore, students are advised to develop homework solutions on their own computer using the JDK, and paste those solutions into JQuiz for submission. During quizzes, use of the JDK is prohibited, and you'll be able to test your code only on JQuiz.

After your work has been graded, you'll be able to review your grade on JQuiz, under the "Grades" tab. Students should carefully review their grade to make sure they understand any applied penalties and to learn from the feedback. In addition, you should review your graded work in a timely manner if you wish to submit a regrade.

## **4 Policies**

### **4.1 Classroom expectations**

Attendance is in-person and mandatory. Remote attendance via Zoom is not offered. Students are expected to arrive to every class promptly: please avoid causing a disruption with a late arrival. You should be actively engaged in the learning process during the duration of the class time.

You are expected to take notes during lecture. Please be prepared to do so.

Active participation in class discussions is strongly encouraged. This is the best time for students to ask questions or clarify any confusing concepts. To participate, please just raise your hand. Students are responsible for any material covered in class, even if it isn't in the textbook. If you miss a class, you should contact a classmate to recover the missed content and assignments.

When in class, please disable or silence any device that may audibly disrupt the class.

During class time, please do not use your phone or computer for activities unrelated to the course. Please do not eat, drink, or sleep during class.

## **4.2 Communication**

Your instructor may use email to make announcements. It is your responsibility to check your Emory email account regularly.

Assignments will typically be posted on Canvas. It is your responsibility to check Canvas for assignments, and to submit your work in a timely manner.

## **4.3 Deadlines, absences, make-ups, and late submissions**

A principle of fairness in evaluation is that all students receive the same opportunity to complete the same work. This principle informs our class policy that the schedule of assignments will not be modified for individual students. In particular:

- All deadlines are strict, final, and automatically enforced. It is the student's responsibility to ensure that they submit their assignments well in advance of the posted deadline. If you miss the deadline for any reason, the submission will not be accepted. Please do not wait until the last minute!
- In general, there is no opportunity for students to make up a missed exam or quiz, or to reschedule an exam or quiz.

However, make-up or rescheduling opportunities may be allowed in certain cases specified by college policy: "A student who fails to take any required midterm or final examination at the scheduled time may not make up the examination without written permission from a dean in the Office for Undergraduate Education. Permission will be granted only for illness or other compelling reasons, such as participation in scheduled events off-campus as an official representative of the University." If this case applies to you, please contact the Office for Undergraduate Education.

## **4.4 Regrades**

You are entitled to a fair and accurate assessment of your work. If you believe that your work has been graded inaccurately, you may request a re-evaluation (henceforth, "regrade") from your professor.

A request for regrade must be submitted to your professor by email within one week of the initial release of the grade. A regrade request must specify the assignment and question to be considered, and must contain an explanation of why the student believes that the grading is inaccurate. Regrade requests are not accepted by means other than email. Regrade requests received after one week will



not be accepted, and at that time all grades are considered final: students are advised to check their grades and review their work promptly. Grades released at the end of the semester may have a shorter window for regrades.

Please note that a regrade may result in a revised grade higher, lower, or the same as the original grade. Submitting regrade requests in bad faith or without appropriate justification will result in a grade penalty.

## 4.5 Getting help

This is a challenging course. If you are struggling with the material, it is your responsibility to take action to improve. There are a variety of ways to get help:

- Ask questions in class. A classroom discussion is the most direct way to understand the material thoroughly.
- Meet with a TA. The teaching assistants offer regular office hours.
- Meet with the instructor. The instructor has regular office hours, as well as office hours by appointment.
- Send an email. You can contact your instructor or a TA by email to ask questions about the homework or course material.
- Read the book. You may prefer the presentation of material in a textual format.
- Ask on the discussion forum. The class site may have an on-line discussion forum where you can post questions.

When asking for help, please consider these guidelines:

- **Don't wait until the last minute.** It's your responsibility to complete assignments on time. Plan accordingly: allow enough time to get any requisite help *before* the deadline.
- **Provide information along with your question.** The person helping you will need to know something about your problem. Be sure to tell them what you are trying to do, what you already did, what happened, and what you expected to happen. Include any relevant code, error messages, or screen shots. Don't just say "It didn't work."
- **Don't expect someone else to do your work for you.** The goal of asking for help is to gain a better understanding of course material. However, when it comes to graded assignments, you still need to do the work for yourself. Expect guidance, not solutions.

## 4.6 Academic integrity

We aim to ensure that your grade reflects your understanding of the material, and therefore we require that all work that you submit, including but not limited to any homework, quiz, exam, or project, be a result solely of your own understanding and effort, except for ungraded in-class exercises, and for those submitted works where collaboration among students is explicitly allowed.

This course is bound by Emory's Honor Code. You are expected to be familiar with this document, particularly the section on academic misconduct. Please also read the Computer Science department's policy on code submission.

### 4.6.1 Motivation

Why do we take academic integrity seriously?

- By committing academic dishonesty, you are cheating yourself out of an opportunity to learn. The challenging work in this course is intended to enhance your skill and understanding. If you bypass that work, you will have failed to learn the material and reduced the value of your college education. Your grade has value only if it accurately represents what you've accomplished to achieve it.
- Academic dishonesty damages the university as an institution. Employers preferentially hire Emory graduates because the Emory degree is conferred on those who have achieved a certain level of education. Acquiring a degree through dishonest means damages the university's reputation and brand.
- Finally, academic dishonesty produces an environment that is unfair to your classmates. If you achieve a grade through dishonest work, students who complete the assignments honestly are at a relative disadvantage.

#### 4.6.2 Policy

We realize that it's not always clear what activities are acceptable in the context of working on your assignments. This is especially true in Computer Science, where the web can provide excellent learning resources (good!) as well as solutions to assignments (bad!).

To help you understand the boundaries of what is and is not acceptable, we provide the following guidelines, which are based on the Stanford CS107 collaboration rules. We classify activities into three categories, according to a traffic light model: those activities that are always acceptable (**green light**), those that are acceptable but require a citation (**orange light**), and those that are never acceptable (**red light**).

##### 4.6.2.1 Acceptable activities

- **Green light** Discussion of general course topics.

You may freely discuss the course material, outside of the context of a specific assignment. This means you can ask and answer questions about the rules of the programming languages and tools that we use, as well as theoretical matters that were covered in lecture or in the textbook. However, the discussion may not refer to solutions for a specific assignment.

For example: "What does the `elem` function do? What is the syntax for a function definition?"

- **Green light** Discussion of assignment requirements.

You may freely discuss the requirements for a specific assignment. However, the discussion may not refer to solutions.

For example: "Are we allowed to use `reverse`? Is performance important?"

- **Green light** Use of public resources for background information.

Web sites, books, and other public resources provide lots of great information. You are expected and encouraged to refer to documentation for the languages and tools that we use, as long as they don't refer to the specific task for your assignment.

For example, the official Python documentation is okay, but "How to write Space Invaders in Python" is not okay (assuming the assignment is to write Space Invaders).

- **Green light** Discussion with teaching staff.

This is the best way to get help! You can always turn to your instructor and TAs with any questions. You can ask questions about any assignment during class, during office hours, or by email.

**4.6.2.2 Activities requiring a citation** What is a citation? If you participate in one of the activities in this section, you must state so in the assignment that it pertains to. If the assignment is a coding exercise, you must write a comment in your code describing the activity. If the assignment is non-code work, you must write a preface describing the activity. Appropriate citation of all external sources is required, including the acknowledgment of any collaboration or assistance.

Every citation must describe (a) the specific source of the information that you used (a person, a book, or a web site), and (b) how you used that information, and to what extent it influenced your work. Failing to provide a citation when participating in one of these activities renders the activity unacceptable, and therefore a violation of academic integrity.

- **Orange light** Re-using your own work.

If you are submitting your own work, including code, that you originally completed before you took this course, you must provide a note stating this fact, including when you originally did the work and under what conditions.

If you want to submit work that you have previously completed for an earlier instance of this course, or for any other course, you must discuss the matter with your instructor before proceeding. In general, you are not allowed to re-submit work completed in an earlier semester.

An exception to this rule applies if you have previously violated academic integrity in this course. In that case, you are prohibited from re-submitting any violating work or any work based on that work.

- **Orange light** Discussion of testing.

If you aren't sure if your solution is correct, it can be helpful to develop a strategy to test it. You may propose and discuss tests collaboratively, as long as you provide a note describing the incident. For example, you may remind a friend that a particular function needs to work with empty lists. However, you may not see each other's code or discuss it in detail.

- **Orange light** Discussion of debugging.

You may discuss particular error messages and propose possible solutions, as long as you provide a note describing the incident. For example, if a friend says they are getting a "segmentation fault," you may suggest that they might be dereferencing a null pointer. However, you may not see each other's code or discuss it in detail.

#### **4.6.2.3 Unacceptable activities**

- **Red light** Copying code or answers from others.

The work you submit must be written entirely by you. This means not only that you physically wrote it, but that the ideas behind it are yours. You may not copy code, read code, or consult code from another person, directly or indirectly. If you are in an environment where you are

able to see another person's solution, it is your responsibility to take yourself out of that environment and actively prevent the possibility of being influenced by their work.

- **Red light** Collaborative development.

You may not work together with another person, and your submitted solution must not be influenced by anyone else's input. Any sharing of technique, design, pseudocode, structure, or substance, even in partial or incomplete form, is not acceptable. Solving bugs collaboratively is prohibited. You must work on your assignments alone, from beginning to end.

- **Red light** Using public resources for assignment-specific code.

Web sites, books, and other public resources can provide valuable information, but any resource that refers to the task that you are to complete for the assignment, or to a task that is substantially similar to it, is not allowed. The design and implementation of your solution may not be influenced or inspired by an external resource. Taking code, in any quantity, from a web site is prohibited. If you find yourself reading a web site that addresses the design or implementation of a program similar to the task at hand, you are required to stop using it.

The boundary between "general background resources" and "assignment-specific resources" may be fuzzy. If you have any doubt that a resource you are using is acceptable, ask your instructor first.

- **Red light** Allowing others to use your code or answers.

At no point should anyone else have access to your work. This means that you may not keep your work on a shared or public computer. You may not post your work on a public web site (including Github and similar sites). You may not show your work to others, even if they promise not to use it in their own work. Your work should remain private, and if at any time it isn't, then you have contributed to academic dishonesty.

You may use Github (and similar sites) as version control storage, but it is your responsibility to ensure that your repository is private. If your code is publicly visible at any time, it constitutes a violation of this policy.

The prohibition on sharing your work remains in effect even after you have submitted your work, after the due date has passed, and after the course has ended. Be aware that other students may attempt to submit your work as their own without your knowledge. Students in future semesters should not have access to your work.

- **Red light** Developing code or solutions with the help of artificial intelligence.

Systems such as Github Copilot and ChatGPT are trained on input data from various sources. As a result, if you use these systems, your program or solution will include elements that you did not write. Therefore, using these or any similar system is prohibited.

- **Red light** Violating the environmental conditions for a quiz or exam.

Quizzes and exams are typically administered in an isolated, proctored environment, without access to external resources. Violating the requirements for the administration of a quiz or exam by, for example, communicating with other people during the period of the quiz or exam, tampering with or bypassing any software tools used in the administration of the quiz or exam,

using prohibited materials during the period of the quiz or exam, or accessing the quiz or exam remotely, represent a violation of academic integrity.

Examination materials are available only to those students taking the quiz or exam during the period of the quiz or exam. Exfiltrating the content of a quiz or exam (for example, by photographing questions) is a violation of academic integrity.

- **Red light** Deliberately providing false information.

Falsifying or failing to provide a required citation is unacceptable. Questions about the provenance of work must be answered honestly. Attempting to deceive the instructor is a violation of course policy.

### 4.6.3 Additional academic integrity policies

If you are in doubt about whether a particular activity is acceptable, it's best to consult with your instructor *before* doing it.

If you are struggling in the course, please discuss your situation with your instructor. Violating academic integrity is not a solution.

Violations of academic integrity will result in immediate referral to the Honor Council. Penalties will depend on the severity of the transgression and each individual student's history of transgressions. Penalties range from a negative score on an assignment or test, failing the course, or even more severe university-wide actions such as suspension or expulsion from the university.

Preventing violations of academic integrity is everyone's responsibility. If you become aware of an instance of academic dishonesty, even if you are not directly involved in it, you are obligated to report it to your instructor.

If you have violated academic integrity, your first step should be to reach out to your instructor so we can work together to find a way for you to move forward with integrity. A timely request to retract a compromised work or apply an amended citation that properly credits authorship could rectify the misrepresentation that would have otherwise been the basis for a strict punishment.

## 4.7 Inclusivity statement

Emory University values an inclusive and equitable environment for all our students. We hope to foster a sense of community in this course and consider it a place where individuals of all backgrounds, beliefs, ethnicities, national origins, gender identities, sexual orientations, religious and political affiliations, and abilities will be treated with respect. All students' learning needs should be addressed both in and out of class, and the diversity that students bring to this course should be viewed as a resource, strength, and benefit. If this standard is not being upheld, please feel free to speak with the instructor.

## 4.8 Academic accommodations

Emory University offers accommodations to students with disabilities. If you anticipate issues related to the format or requirements of this course, please meet with the instructor to discuss ways to ensure your full participation in the course. If you determine that disability related accommodations are necessary, please register with the Department of Accessibility Services as soon as possible.

Students seeking academic accommodations for religious observance should submit their requests to the instructor as early as possible in advance. If you have questions or concerns about your request, you may contact the university's Office of Spiritual and Religious Life (OSRL), the Ombuds Office, or the Office of Institutional Equity and Compliance (OIEC). Academic accommodations for religious observance do not relieve students of responsibility for the completion of any part of the coursework they may miss as the result of a religious observance.