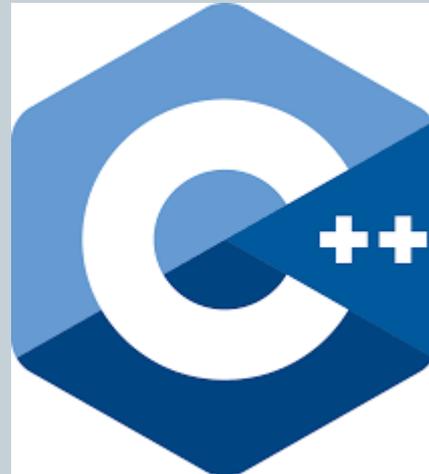


Course Overview, Python Review, API, Comments



Seating Chart

Door

**S&E Room: 308
Computer Lab**

**Instructor:
Michele Van Dyne**

CLASS: _CSCI 136
TIME: _11:00-11:50, 2:00-4:50_
DAYS: M T W TH F

Door

	Brayden Erfle		Adel Bigart		Will Augustine		
COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER		
	↕		↕		↕		
Danielle Failor	William Franzen	Clay Fulk	Roland Grena	Seth Gutierrez	Erin Hennelly	Ty Insko	Alexus Jenkins
↔	↔	↔	↔	↔	↔	↔	↔
COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER
Reese Lester	Dylan Mentzer	Michael Nelson	Spencer Reitz	Jace Rhodes	Jacob Richardson	Shelby Schweigert	River Sheppard
↔	↔	↔	↔	↔	↔	↔	↔
COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER
	↕		↕		↕		↕
	Mason Simon		Jessa Steele		Zachary Tomaszewski		Hunter Valdez
COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER

**Instructor's
Computer**

TABLE

Door

Outline

- **Course Overview**
 - Website
 - Syllabus
 - Assignments
 - Exams
- **Python Review**
- **API**
- **Comments**

Course Website and Other Important Info

Michele Van Dyne

Museum 204B

Office Hours: MWF 1:00-2:00
or by appt.

mvandyne@mtech.edu

Course Website:

<https://katie.mtech.edu/classes/csci136/>

Moodle:

- Use to submit assignments and to check your grades

Topics and Schedule

CSCI 136
Fundamentals of Computer Science II
Spring 2021

MontanaTech

[Computer Science & Software Engineering](#)

[Schedule](#) | [Assignments](#) | [Resources](#) | [Course Syllabus](#) | [Moodle](#)

SCHEDULE

This page lists the anticipated dates of all the lectures with links to slides, web references and examples from the lecture (if any). Readings are posted for chapters in books available through the library eBook portal.

#	Date	Topic	Slides	Resources	Examples
1	Mon. 1/4	Course Overview API Review		Link to CSCI 135 Materials	
-	Wed. 1/6	Lab 1: Python Review			
2	Fri. 1/8	Dynamic Arrays			
3	Mon. 1/11	Linked Lists			
-	Wed. 1/13	Lab 2: Linked Lists			
4	Fri. 1/15	Stacks and Queues			
-	Mon. 1/18	NO CLASS - Martin Luther King Day			
-	Wed. 1/20	Lab 3: Stacks and Queues			
5	Fri. 1/22	Abstract Data Types, Interfaces			
6	Mon. 1/25	Hash Maps, Performance			
-	Wed. 1/27	Lab 4: Abstract Classes			
7	Fri. 1/29	Recursion I			
8	Mon. 2/1	Recursion II			
-	Wed. 2/3	Lab 5: Recursion			
9	Fri. 2/5	Fun with Recursion			
-	Mon. 2/8	Exam Review			
-	Wed. 2/10	EXAM 1			
10	Fri. 2/12	Threads			

Lab Assignments

CSCI 136
Fundamentals of Computer Science II
Spring 2021

MontanaTech

[Computer Science & Software Engineering](#)

[Schedule](#) | [Assignments](#) | [Resources](#) | [Course Syllabus](#) | [Moodle](#)

SCHEDULE

This page lists the anticipated dates of all the lectures with links to slides, web references and examples from the lecture (if any). Readings are posted for chapters in books available through the library eBook portal.

#	Date	Topic	Slides	Resources	Examples
1	Mon. 1/4	Course Overview API Review		Link to CSCI 135 Materials	
-	Wed. 1/6	Lab 1: Python Review			
2	Fri. 1/8	Dynamic Arrays			
3	Mon. 1/11	Linked Lists			
-	Wed. 1/13	Lab 2: Linked Lists			
4	Fri. 1/15	Stacks and Queues			
-	Mon. 1/18	NO CLASS - Martin Luther King Day			
-	Wed. 1/20	Lab 3: Stacks and Queues			
5	Fri. 1/22	Abstract Data Types, Interfaces			
6	Mon. 1/25	Hash Maps, Performance			
-	Wed. 1/27	Lab 4: Abstract Classes			
7	Fri. 1/29	Recursion I			
8	Mon. 2/1	Recursion II			
-	Wed. 2/3	Lab 5: Recursion			
9	Fri. 2/5	Fun with Recursion			
-	Mon. 2/8	Exam Review			
-	Wed. 2/10	EXAM 1			
10	Fri. 2/12	Threads			

Resources

CSCI 136
Fundamentals of Computer Science II
Spring 2021

MontanaTech

[Computer Science & Software Engineering](#)

[Schedule](#) | [Assignments](#) | [Resources](#) | [Course Syllabus](#) | [Moodle](#)

SCHEDULE

This page lists the anticipated dates of all the lectures with links to slides, web references and examples from the lecture (if any). Readings are posted for chapters in books available through the library eBook portal.

#	Date	Topic	Slides	Resources	Examples
1	Mon. 1/4	Course Overview API Review		Link to CSCI 135 Materials	
-	Wed. 1/6	Lab 1: Python Review			
2	Fri. 1/8	Dynamic Arrays			
3	Mon. 1/11	Linked Lists			
-	Wed. 1/13	Lab 2: Linked Lists			
4	Fri. 1/15	Stacks and Queues			
-	Mon. 1/18	NO CLASS - Martin Luther King Day			
-	Wed. 1/20	Lab 3: Stacks and Queues			
5	Fri. 1/22	Abstract Data Types, Interfaces			
6	Mon. 1/25	Hash Maps, Performance			
-	Wed. 1/27	Lab 4: Abstract Classes			
7	Fri. 1/29	Recursion I			
8	Mon. 2/1	Recursion II			
-	Wed. 2/3	Lab 5: Recursion			
9	Fri. 2/5	Fun with Recursion			
-	Mon. 2/8	Exam Review			
-	Wed. 2/10	EXAM 1			
10	Fri. 2/12	Threads			

Syllabus

CSCI 136
Fundamentals of Computer Science II
Spring 2021

MontanaTech

[Computer Science & Software Engineering](#)

[Schedule](#) | [Assignments](#) | [Resources](#) | [Course Syllabus](#) | [Moodle](#)

SCHEDULE

This page lists the anticipated dates of all the lectures with links to slides, web references and examples from the lecture (if any). Readings are posted for chapters in books available through the library eBook portal.

#	Date	Topic	Slides	Resources	Examples
1	Mon. 1/4	Course Overview API Review		Link to CSCI 135 Materials	
-	Wed. 1/6	Lab 1: Python Review			
2	Fri. 1/8	Dynamic Arrays			
3	Mon. 1/11	Linked Lists			
-	Wed. 1/13	Lab 2: Linked Lists			
4	Fri. 1/15	Stacks and Queues			
-	Mon. 1/18	NO CLASS - Martin Luther King Day			
-	Wed. 1/20	Lab 3: Stacks and Queues			
5	Fri. 1/22	Abstract Data Types, Interfaces			
6	Mon. 1/25	Hash Maps, Performance			
-	Wed. 1/27	Lab 4: Abstract Classes			
7	Fri. 1/29	Recursion I			
8	Mon. 2/1	Recursion II			
-	Wed. 2/3	Lab 5: Recursion			
9	Fri. 2/5	Fun with Recursion			
-	Mon. 2/8	Exam Review			
-	Wed. 2/10	EXAM 1			
10	Fri. 2/12	Threads			

Moodle

CSCI 136
Fundamentals of Computer Science II
Spring 2021

MontanaTech

[Computer Science & Software Engineering](#)

[Schedule](#) | [Assignments](#) | [Resources](#) | [Course Syllabus](#) | [Moodle](#)

SCHEDULE

This page lists the anticipated dates of all the lectures with links to slides, web references and examples from the lecture (if any). Readings are posted for chapters in books available through the library eBook portal.

#	Date	Topic	Slides	Resources	Examples
1	Mon. 1/4	Course Overview API Review		Link to CSCI 135 Materials	
-	Wed. 1/6	Lab 1: Python Review			
2	Fri. 1/8	Dynamic Arrays			
3	Mon. 1/11	Linked Lists			
-	Wed. 1/13	Lab 2: Linked Lists			
4	Fri. 1/15	Stacks and Queues			
-	Mon. 1/18	NO CLASS - Martin Luther King Day			
-	Wed. 1/20	Lab 3: Stacks and Queues			
5	Fri. 1/22	Abstract Data Types, Interfaces			
6	Mon. 1/25	Hash Maps, Performance			
-	Wed. 1/27	Lab 4: Abstract Classes			
7	Fri. 1/29	Recursion I			
8	Mon. 2/1	Recursion II			
-	Wed. 2/3	Lab 5: Recursion			
9	Fri. 2/5	Fun with Recursion			
-	Mon. 2/8	Exam Review			
-	Wed. 2/10	EXAM 1			
10	Fri. 2/12	Threads			

Getting Help



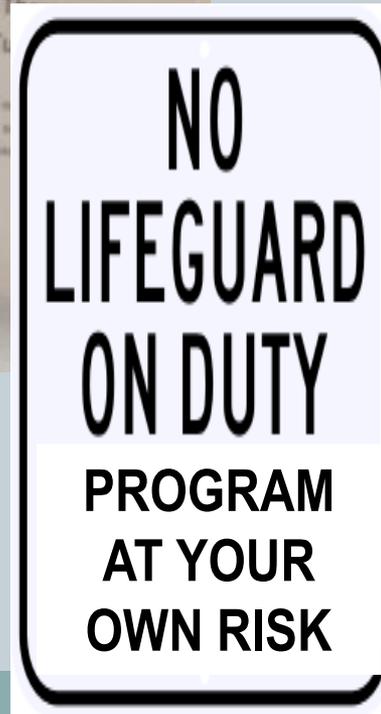
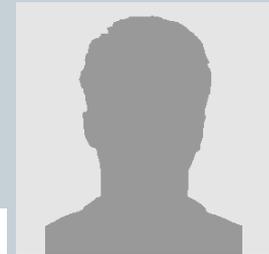
Michele Van Dyne
Museum 204B
mvandyne@mtech.edu

Office hours:

Mon., Wed., Fri., 1-2
or by appointment



We will have at least one
teaching assistant:



How to Ace This Class

- Come to lecture
- Do the in-class extra credit assignments
- Complete all the labs
 - Even if you are not done, turn them in – partial credit helps
 - Save your code on a USB drive or cloud drive or email it to yourself – files can disappear
- Experiment!
 - Try the code examples in the readings, download the posted examples for class and change them, see what happens
- If you have a question in class – ask!
 - If you have a question, it's likely that others do also
 - And sometimes... I may not explain something clearly – tell me!
- If you have a question outside of class
 - Talk to the lab tutors in the Museum Computer Lab
 - Email me
 - Use Discord to ask questions
- Don't copy code... from someone else or from the internet

Python Review

- CSCI 135 Materials from last semester available at:
 - <https://katie.mtech.edu/classes/csci135/>
 - ✦ Variables, Data Types, Expressions
 - ✦ Lists
 - ✦ Conditionals
 - ✦ Iteration
 - ✦ Functions
 - ✦ File I/O
 - ✦ Classes, Methods, Attributes
 - ✦ Exceptions and Testing

API

- Application Programming Interface
 - Name of a method or function
 - Parameters it requires (and their data types)
 - Return value(s) (and their data types)
 - Description of what the method or function does

```
class Balloon
```

<u>Return:</u>	<u>Method:</u>	<u>Parameters:</u>
	Balloon	(self, float x, float y, float <u>maxRadius</u> , string label, int red, int green, int blue)
	<u>draw</u>	(self)
boolean	<u>matches</u>	(self, string label)
float	<u>lift</u>	(self, float <u>liftPerUnitVolume</u>)
	<u>inflate</u>	(self, float <u>increaseRadiusBy</u>)

__main__

- You've seen a test main in some of our examples and programs before

```
#
# Test program for Cell.
#
# To run this code and test your class, type "python Cell.py" at the
# command prompt. Your output should be exactly like what is shown
# on the assignment page.
#
if __name__ == "__main__":
    import Config
    # The config object stores defaults for sizes, sounds, and images
    config = Config.Config()

    # Fake a dungeon that is 3 x 3
    WIDTH = 3
    HEIGHT = 3
    StdDraw.setCanvasSize(WIDTH * config.cellPixels(), HEIGHT * config.cellPixels())
    StdDraw.setXscale(0, WIDTH * config.cellPixels())
    StdDraw.setYscale(0, HEIGHT * config.cellPixels())

    # Draw a wall at various locations
    wall = Cell(config.wallImage())
    wall.draw(config, 0, 0)
    wall.draw(config, 1, 1)

    # Wall with a different image
    wall2 = Cell(config.passageImage())
    wall2.draw(config, 2, 2)
```

New! Autograder

- I will attempt to write an Autograder program for as many of your programs as possible, and make this available to you along with weekly lab assignments
 - You can run this to make sure your program is running as expected
 - You can get an unbiased score for the executable points in your program
 - ✦ Other points will still have to be determined by hand
 - Comments, style, etc.

Comments

- **REQUIRED** for your code this semester:
 - As always, header comment with your name and a description of the code
 - Comments before each method or function, describing the purpose of the code block
 - ✦ **Additionally:**
 - Include the input parameters (name, data type expected, and purpose)
 - Include the return value(s) – if any – and their data type and purpose
 - Use additional comments to point out tricky parts of the code

Summary

- **Course Overview**
 - Website
 - Syllabus
 - Assignments
 - Exams
- **Python Review**
- **API**
- **Comments**

