Course Mechanics
Outline

1. Course Logistics

2. Course Overview
Course Logistics

Course website: https://www.cs.umb.edu/~msolah/cs110_f18/index.html

Goal: proficiency in the design and implementation of Python programs of significant size and complexity

Prerequisites: Math 140 credits or placement; or Math 130 as corequisite; or permission of the department

Instructor: Michael Solah

Twice a week classes

Weekly discussion sessions (start second week)

Weekly supplemental instruction sessions (start second week)

Tutoring

Text: Introduction to Programming in Python – An Interdisciplinary Approach by Robert Sedgewick et al
Course Logistics

Course website: https://www.cs.umb.edu/~msolah/cs110_f18/index.html

Goal: proficiency in the design and implementation of Python programs of significant size and complexity

Prerequisites: Math 140 credits or placement; or Math 130 as corequisite; or permission of the department

Instructor: Michael Solah

Twice a week classes

Weekly discussion sessions (start second week)

Weekly supplemental instruction sessions (start second week)

Tutoring

Text: Introduction to Programming in Python – An Interdisciplinary Approach by Robert Sedgewick et al
Course Logistics

Course website: https://www.cs.umb.edu/~msolah/cs110_f18/index.html

Goal: proficiency in the design and implementation of Python programs of significant size and complexity

Prerequisites: Math 140 credits or placement; or Math 130 as corequisite; or permission of the department

Instructor: Michael Solah

Twice a week classes

Weekly discussion sessions (start second week)

Weekly supplemental instruction sessions (start second week)

Tutoring

Text: Introduction to Programming in Python – An Interdisciplinary Approach by Robert Sedgewick et al
Course Logistics

Course website: https://www.cs.umb.edu/~msolah/cs110_f18/index.html

Goal: proficiency in the design and implementation of Python programs of significant size and complexity

Prerequisites: Math 140 credits or placement; or Math 130 as corequisite; or permission of the department

Instructor: Michael Solah

Twice a week classes

Weekly discussion sessions (start second week)

Weekly supplemental instruction sessions (start second week)

Tutoring

Text: Introduction to Programming in Python – An Interdisciplinary Approach by Robert Sedgewick et al
Course Logistics

Course website: https://www.cs.umb.edu/~msolah/cs110_f18/index.html

Goal: proficiency in the design and implementation of Python programs of significant size and complexity

Prerequisites: Math 140 credits or placement; or Math 130 as corequisite; or permission of the department

Instructor: Michael Solah

Twice a week classes

Weekly discussion sessions (start second week)

Weekly supplemental instruction sessions (start second week)

Tutoring

Text: Introduction to Programming in Python – An Interdisciplinary Approach by Robert Sedgewick et al
Course Logistics

Course website: https://www.cs.umb.edu/~msolah/cs110_f18/index.html

Goal: proficiency in the design and implementation of Python programs of significant size and complexity

Prerequisites: Math 140 credits or placement; or Math 130 as corequisite; or permission of the department

Instructor: Michael Solah

Twice a week classes

Weekly discussion sessions (start second week)

Weekly supplemental instruction sessions (start second week)

Tutoring

Text: Introduction to Programming in Python – An Interdisciplinary Approach by Robert Sedgewick et al
Course Logistics

Course website: https://www.cs.umb.edu/~msolah/cs110_f18/index.html

Goal: proficiency in the design and implementation of Python programs of significant size and complexity

Prerequisites: Math 140 credits or placement; or Math 130 as corequisite; or permission of the department

Instructor: Michael Solah

Twice a week classes

Weekly discussion sessions (start second week)

Weekly supplemental instruction sessions (start second week)

Tutoring

Text: Introduction to Programming in Python – An Interdisciplinary Approach by Robert Sedgewick et al
Course Logistics

Course website: https://www.cs.umb.edu/~msolah/cs110_f18/index.html

Goal: proficiency in the design and implementation of Python programs of significant size and complexity

Prerequisites: Math 140 credits or placement; or Math 130 as corequisite; or permission of the department

Instructor: Michael Solah

Twice a week classes

Weekly discussion sessions (start second week)

Weekly supplemental instruction sessions (start second week)

Tutoring

Text: Introduction to Programming in Python – An Interdisciplinary Approach by Robert Sedgewick et al
Course Logistics

Course website: https://www.cs.umb.edu/~msolah/cs110_f18/index.html

Goal: proficiency in the design and implementation of Python programs of significant size and complexity

Prerequisites: Math 140 credits or placement; or Math 130 as corequisite; or permission of the department

Instructor: Michael Solah

Twice a week classes

Weekly discussion sessions (start second week)

Weekly supplemental instruction sessions (start second week)

Tutoring

Text: Introduction to Programming in Python – An Interdisciplinary Approach by Robert Sedgewick et al
Course Logistics

Course website: https://www.cs.umb.edu/~msolah/cs110_f18/index.html

Goal: proficiency in the design and implementation of Python programs of significant size and complexity

Prerequisites: Math 140 credits or placement; or Math 130 as corequisite; or permission of the department

Instructor: Michael Solah

Twice a week classes

Weekly discussion sessions (start second week)

Weekly supplemental instruction sessions (start second week)

Tutoring

Text: Introduction to Programming in Python – An Interdisciplinary Approach by Robert Sedgewick et al
Course Logistics

Grading

• Homework assignments (best 8 of 9): 8%
• Project assignments (best 5 of 6): 30%
• Exams (best 2 of 3): 50%
• Attendance & Participation: 12%

Attendance

Piazza (Q&A forum)

Gradescope (grading platform)

CS account

Programming environment (a Linux-based virtual machine)

Policies

• Classroom
• Piazza
• Collaboration
• Code of conduct
• Accommodations for students with disabilities
Course Logistics

Grading

• Homework assignments (best 8 of 9): 8%
• Project assignments (best 5 of 6): 30%
• Exams (best 2 of 3): 50%
• Attendance & Participation: 12%

Attendance

Piazza (Q&A forum)

Gradescope (grading platform)

CS account

Programming environment (a Linux-based virtual machine)

Policies

• Classroom
• Piazza
• Collaboration
• Code of conduct
• Accommodations for students with disabilities
Course Logistics

Grading

- Homework assignments (best 8 of 9): 8%
- Project assignments (best 5 of 6): 30%
- Exams (best 2 of 3): 50%
- Attendance & Participation: 12%

Attendance

Piazza (Q&A forum)

Gradescope (grading platform)

CS account

Programming environment (a Linux-based virtual machine)

Policies

- Classroom
- Piazza
- Collaboration
- Code of conduct
- Accommodations for students with disabilities
Course Logistics

Grading

• Homework assignments (best 8 of 9): 8%
• Project assignments (best 5 of 6): 30%
• Exams (best 2 of 3): 50%
• Attendance & Participation: 12%

Attendance

Piazza (Q&A forum)

Gradescope (grading platform)

CS account

Programming environment (a Linux-based virtual machine)

Policies

• Classroom
• Piazza
• Collaboration
• Code of conduct
• Accommodations for students with disabilities
Course Logistics

Grading

- Homework assignments (best 8 of 9): 8%
- Project assignments (best 5 of 6): 30%
- Exams (best 2 of 3): 50%
- Attendance & Participation: 12%

Attendance

Piazza (Q&A forum)

Gradescope (grading platform)

CS account

Programming environment (a Linux-based virtual machine)

Policies

- Classroom
- Piazza
- Collaboration
- Code of conduct
- Accommodations for students with disabilities
Course Logistics

Grading

- Homework assignments (best 8 of 9): 8%
- Project assignments (best 5 of 6): 30%
- Exams (best 2 of 3): 50%
- Attendance & Participation: 12%

Attendance

Piazza (Q&A forum)

Gradescope (grading platform)

CS account

Programming environment (a Linux-based virtual machine)

Policies

- Classroom
- Piazza
- Collaboration
- Code of conduct
- Accommodations for students with disabilities
Course Logistics

Grading

• Homework assignments (best 8 of 9): 8%
• Project assignments (best 5 of 6): 30%
• Exams (best 2 of 3): 50%
• Attendance & Participation: 12%

Attendance

Piazza (Q&A forum)

Gradescope (grading platform)

CS account

Programming environment (a Linux-based virtual machine)

Policies

• Classroom
• Piazza
• Collaboration
• Code of conduct
• Accommodations for students with disabilities
Course Logistics

Grading

• Homework assignments (best 8 of 9): 8%
• Project assignments (best 5 of 6): 30%
• Exams (best 2 of 3): 50%
• Attendance & Participation: 12%

Attendance

Piazza (Q&A forum)

Gradescope (grading platform)

CS account

Programming environment (a Linux-based virtual machine)

Policies

• Classroom
• Piazza
• Collaboration
• Code of conduct
• Accommodations for students with disabilities
Course Logistics

Tips to succeed

Items on the course website
- Announcements (landing page)
- Course Info
- Calendar
- Slides
- Assignments
- Resources

Things to do immediately
- Sign up for Piazza
- Sign up for Gradescope
- Apply for a CS account
- Setup the programming environment
Course Logistics

Tips to succeed

Items on the course website
- Announcements (landing page)
- Course Info
- Calendar
- Slides
- Assignments
- Resources

Things to do immediately
- Sign up for Piazza
- Sign up for Gradescope
- Apply for a CS account
- Setup the programming environment
Course Logistics

Tips to succeed

Items on the course website
- Announcements (landing page)
- Course Info
- Calendar
- Slides
- Assignments
- Resources

Things to do immediately
- Sign up for Piazza
- Sign up for Gradescope
- Apply for a CS account
- Setup the programming environment
Course Logistics

Tips to succeed

Items on the course website
• Announcements (landing page)
• Course Info
• Calendar
• Slides
• Assignments
• Resources

Things to do immediately
• Sign up for Piazza
• Sign up for Gradescope
• Apply for a CS account
• Setup the programming environment
Course Overview

Introduction
• What is Computer Science?
• Building a Computer

Imperative Programming
• Your First Program
• Built-in Types of Data
• Control Flow
• Collections
• Input and Output
• Case Study: What Makes Google Different? (PageRank Algorithm)
Course Overview

Introduction
  • What is Computer Science?
  • Building a Computer

Imperative Programming
  • Your First Program
  • Built-in Types of Data
  • Control Flow
  • Collections
  • Input and Output
  • Case Study: What Makes Google Different? (PageRank Algorithm)
Course Overview

Introduction
• What is Computer Science?
• Building a Computer

Imperative Programming
• Your First Program
• Built-in Types of Data
• Control Flow
• Collections
• Input and Output
• Case Study: What Makes Google Different? (PageRank Algorithm)
Course Overview

Procedural Programming
- Defining Functions
- Modules and Clients
- Recursion
- Case Study: Fermi’s Paradox (Percolation Problem)

Object-oriented Programming
- Using Data Types
- Creating Data Types
- Designing Data Types
- Case Study: The Music of the Spheres (N-body Problem)

Algorithms and Data Structures
- Searching and Sorting
- Stacks and Queues
- Case Study: Six Degrees of Separation (Small-world Problem)

Limits of Computation
Course Overview

Procedural Programming
• Defining Functions
• Modules and Clients
• Recursion
• Case Study: Fermi’s Paradox (Percolation Problem)

Object-oriented Programming
• Using Data Types
• Creating Data Types
• Designing Data Types
• Case Study: The Music of the Spheres (N-body Problem)

Algorithms and Data Structures
• Searching and Sorting
• Stacks and Queues
• Case Study: Six Degrees of Separation (Small-world Problem)

Limits of Computation
Course Overview

Procedural Programming
• Defining Functions
• Modules and Clients
• Recursion
• Case Study: Fermi’s Paradox (Percolation Problem)

Object-oriented Programming
• Using Data Types
• Creating Data Types
• Designing Data Types
• Case Study: The Music of the Spheres (N-body Problem)

Algorithms and Data Structures
• Searching and Sorting
• Stacks and Queues
• Case Study: Six Degrees of Separation (Small-world Problem)

Limits of Computation
Course Overview

Procedural Programming
- Defining Functions
- Modules and Clients
- Recursion
- Case Study: Fermi’s Paradox (Percolation Problem)

Object-oriented Programming
- Using Data Types
- Creating Data Types
- Designing Data Types
- Case Study: The Music of the Spheres (N-body Problem)

Algorithms and Data Structures
- Searching and Sorting
- Stacks and Queues
- Case Study: Six Degrees of Separation (Small-world Problem)

Limits of Computation
Course Overview

Procedural Programming
- Defining Functions
- Modules and Clients
- Recursion
- Case Study: Fermi’s Paradox (Percolation Problem)

Object-oriented Programming
- Using Data Types
- Creating Data Types
- Designing Data Types
- Case Study: The Music of the Spheres ($N$-body Problem)

Algorithms and Data Structures
- Searching and Sorting
- Stacks and Queues
- Case Study: Six Degrees of Separation (Small-world Problem)

Limits of Computation