Visiting Assoc. Prof. Mihaela Malita mmalita@amherst.edu
Amherst College. Computer Science Department, Science Center Room **C217** *Phone: 413-542.8512*
Office Hours: **Th @11:00 AM – 11:50 AM, @ 4:00 – 6:00 PM and anytime we agree upon**

COSC 112-01 Introduction to Computer Science II. Fall 2021

Syllabus

Lectures: MW @11:00 – 11:50 AM in Room A131

Labs: Friday @10:00 – 10:50 AM, @11:00–11:50 AM in Room A331

Course website: **https://mmalita.people.amherst.edu/COSC112FA21**

It does not matter how slowly you go so long as you do not stop. Confucius

---

Credits: 4

Prerequisite: *COSC 111 Intro to Computer Science I*

Required for: For the CS Major. For Major in Statistics (Satisfy a computing requirement) see https://www.amherst.edu/academiclife/departments/mathematics-statistics/major-in-statistics/pathways-through-the-major

Course has a LAB component.

Course Overview (Catalog):

A continuation of COSC 111. This course will emphasize more complicated problems and their algorithmic solutions. The object-oriented programming paradigm will be discussed in detail, including data abstraction, inheritance, and polymorphism. Other topics will include stacks, queues, linked lists, programming for graphical user interfaces, and basic topics in probability. A laboratory section will meet once a week to give students practice with programming constructs.

Textbooks:

- Lyle McGeoch, *Java Programming, 2012* (PDF only, on Moodle)
- Cutajar, James. "Beginning Java Data Structures and Algorithms [ebook] Packt Publishing, 2019 (on Moodle)
- Stuart Reges,Marty Stepp, Building Java Programs: A Back to Basics Approach, 5th ed, Pearson, 2019, https://www.buildingjavaprograms.com/youtube.shtml. Chapter gui-sample-chapter.pdf (Moodle)
- Robert Sedgewick, Kevin Wayne, Intro to Programming in Java: An interdisciplinary approach, 2ed. StdDraw Or Computer Science An Interdisciplinary Approach, Part I, Princeton Univ, 2019

Software used:

- **J a v a** download free: https://www.java.com/en/ || Java Shell on line: https://tryjshell.org/
- Google Drive: COSC112FA21 here are files and solutions
- **Moodle:** all course materials and points
- **Slack:** https://amherstcollege.slack.com for communication

Amherst IT support: https://www.amherst.edu/offices/it/services/help/help-desk

College Calendar. Fall 2021: Grades are due on Tuesday, December 28

Mon, Aug 30	Fall Classes Begin, First Day of Add/Drop
Wed Sept 8	Last Day of Add/Drop and Pass/Fail, End of 5-College Interchange Registration
Sat, Oct 9 until Oct 12	**Mid-Semester Break**
Wed, Oct 13	Fall Classes Resume, MON, OCT 25 UNTIL OCT 29 Advising Week,
Th, Oct 15	Deadline for mid-semester warning grades (D/F)
Mon, Nov 1 until Nov 7	Pre-Registration for Spring 2022
Mon, Nov 8 until Nov 12	Roster Management Week

Mon, Nov 15 until Nov 19	Second Round Registration Week
Sat, Nov 20 until Nov 28	**Thanksgiving Break**
Tue, Dec 7	Last Day of Classes and Last Day for Major Declaration Class of 24E
Wed, Dec 8 until Dec 12	Reading/Study Period
Mon, Dec 13 until Dec 17	**Final Exam Week**

Course Calendar Fall 2021:

Week #	Date	Handout
1	Mon 08/30	Week#1 + Lab1
2	Mon 09/06	Week#2 + Lab2 + **Hwk2 [3 p]** due Fr 09/17
3	Mon 09/13	Week#3 + Lab3 + **Hwk3 [3 p]** due Fr 09/24
4	Mon 09/20	Week#4 + Lab4 + **Hwk4 [3 p]** due Fr 10/01
5	Mon 09/27	Week#5 + Lab5
6	Mon 10/04	Week#6 + Lab6: Review midterm
7	**Mon 10/11(BREAK).** Wed 10/13	Week #7 + Lab7: MIDTERM [25 p] is 10/13
8	Mon 10/18	Week #8 + **Hwk8 [3 p]** due Fr 10/29
9	Mon 10/25	Week #9 + Lab9 + **Hwk9 [3 p]** due Fr 11/05
10	Mon 11/01	Week #10 + Lab10 + **Hwk10 [3 p]** due Fr 11/12
11	Mon 11/08	Week #11 + Lab11
12	Mon 11/15	Week #12 + Lab12: Work on Project
13	**Mon 11/22- Fr 11/26**	**Thanksgiving BREAK**
14	Mon 11/29	Week #14 + Lab14: **Project Presentation [20 p]**
15	Mon 12/06 last lecture	Week #15 + NO Lab15 + Review for Final
16	Mon 12/13 - Fr 12/17	Final Exam [32 p]

List of Topics Covered

- Defining and creating object classes, object allocation and garbage collection
- Class hierarchy, Inheritance, overloading and overriding
- Abstract classes and methods, Interfaces
- Generics and container classes
- Linked lists, Stacks, Queues
- GUI interfaces and event-driven programming

Teaching method: Active learning

An algorithm must be seen to be believed. Donald Knuth

As I type programs on my computer, I will explain their meaning. Students will be asked to reproduce immediately the code that I have written. I explain the new concepts by showing them how and where to use them. Frequently questions are asked as students can jump in anytime with their questions and can also propose alternative solutions. Students are asked to build a **portfolio** with all the programs they write during the semester posted in their Google drive. Students will always have a **different** partner for every homework and might work with a partner for their final project. Homework solutions are discussed during lab time and all projects will be presented in our final day of class.

Learning Outcomes:
By the end of this course the student should be able to do the following:

- Design, implement, test and document a program in Java
- Know how to start from a problem statement and build a program to solve the problem
- Demonstrate understanding of b a s i c data structures for use in any CS application
- Learn how to work from your terminal. Master your own computer
- Basic Math concepts used in CS: sets, sequences, combinatorics, linear algebra
- Present your software application in front of an audience and work in a group

Assessment has 4 parts (total 100 points)

1. **Lab [5 points].** There is a Lab each week (13 Labs) in which students are assigned 6-25 questions / programming problems. If they are unable to finish them during lab time they have to submit them as extra homework. Lab problems will be graded at the end of the semester when students will show their Lab portfolio.
    - 0 Points – less than 20% submitted + no participation
    - 1 Points – Lab problems submitted 40% + no participation
    - 2 Points – Lab problems submitted 50% + no participation
    - 3 Points – Lab problems submitted 60 % + participation
    - 4 Points – Lab problems finished 80 % + participation
    - 5 Points – Lab problems finished 90 % + participation + helped others
2. **Midterm Exam (50 mins in class) [25 points]**. Some questions theoretical questions and short program in Java to write.
3. **Homework assignments [18 points]**. Students will be assigned 6 hwks (3 points each) consisting of 1-5 Java programs. The assignments will be based on their work done in that week's lectures and lab. In addition, a number of readings and short movies will be assigned during the semester. Preferably all home works will be done with a **different** partner(s). Students will keep track of their partners on our class Google drive spreadsheet: Google/**partners112.xlsx**. The students in a team can have the same program or different programs (they will consult with each other and help debugging), it is their choice, but each of them has to be submitted separate in Moodle as their homework-programs.
    - 0 Points - nothing submitted
    - 1 Points - program submitted, but incomplete
    - 2 Points - program submitted, works, but poor design, no comments
    - 3 Points – very well

4. **A Final Project[20 points]** will be assigned which consists of a program written in Java that incorporates all the concepts students have learned in class (a **data base** type of problem: a collection of objects (data) to be read from a **file**(s) and reorganize (sort after different criteria), retrieve, modify, process and display and report in a file. The program should be structured (have a **menu**), have at least 2 **object**s and be properly commented. All projects should contain a **readme.txt** file presenting title, authors, date, course, files needed in the program and a short abstract of what is the purpose of the program. Final Project is 20 points. All projects will be presented in class during our last labs/classes. Topic can be chosen from another class. All topics must be approved by the instructor. You will work with a partner of your choice. Here are some project titles from previous years:
Adventure Games, Adventure through mini games, Attendance Checker, Baccarat, Bank Manager, Battle Ship Game, Boston Bruins Stats, Car stats, Casino Games, Countries Stats, CS Job Salaries, Database Arab

revolts, Dog Breeds, Event Finder, Fictional Character Database and Randomizer, Food Order App, Fortune Teller, Go Fish, Hangman, Kpop Music Library, Mastermind, Minesweeper, Movie Library, Puerto Rico Governors, QUIXO, Quiz Builder, Scrabble Game, Shelter Search, Soccer Stats, Store Management, Sudoku, Time Manager, Todo Helper, Vacation App, Vocab FlashCard Building, Voting, Who wants to be a millionaire?, Wrath of Manasses, Yahtzee.

You are encouraged to choose a topic of your interest that might also be inspired from another class.

Midterm[**25**], Final Exam[**32**] ,Home works [**18]**, Final Project [**20**], Lab[**5**] = Total [**100** points]

Grade Range (in points):

A 95-100; **A-** 90-94;

B+ 87-89; **B** 83-86; **B-** 80-82;

C+ 77-79; **C** 73-76; **C-** 70-72;

D+ 67-69; **D** 60-66; **F <** 60

Examination Policy. The final exam is comprehensive and will cover all the material from the semester. Any homework, test, project can be followed by questions from the teacher.

Lateness Policy. Except for illness or very special circumstances, no late days will be granted for home works and projects. Class and lab attendance is required and will count towards the final grade. All course work in a given semester must be submitted by the last day of classes at 5:00 p.m.

Expected Work. Follow rules from "Student Code of Conduct":

https://www.amherst.edu/offices/student-affairs/community-standards/student-code-of-conduct

Writing component. Students should comment all their programs. Students will be asked to write short notes on significant portraits from math / computer science and on the history of Java and object oriented programming. They will also explain in detail their project and follow the rules regarding how to write documentation and use them properly for the final project documentation.