# CSCI 3920: Advanced Programming with Java and Python
# Course Syllabus
# Fall, 2021


## General Information

**Instructor:** Javier Pastorino
**Email:** javier.pastorino@ucdenver.edu
**Office:** Virtual Office
**Office Hours:** Schedule Appointment Through Calendly.

## Course Description

### Catalog Data

This course introduces the fundamental concepts to develop programs and projects using modern software engineering techniques and two different programming languages (Java and Python). It will cover and apply pattern design approaches, reusable components driven by everyday needs within many software developments, the relationships between object-oriented programming concepts and software design concepts. It will dig deeper into techniques to program single threaded applications as well as advanced techniques to construct concurrent and distributed applications.

### Course Objectives

Designing and developing software has many aspects. Many of these aspects are best approached without a specific language in mind. The concepts behind most Software Engineering, Program Design, and Program Construction techniques are language independent.
The goal of this course is to introduce two new languages, one fully Object-Oriented and another one that follows either Procedural or Object-Oriented paradigms. This is done while engaging the students in advanced programming techniques like multithreading and concurrent programming as well as introducing Software Design practices.

## Course Format

This course will be given in a hybrid setting. A **remote asynchronous setting** will be used for lectures, homework, and quizzes. Pre-recorded lecture videos will be posted weekly, and online collaboration tools such as Canvas and Slack/Discord chat will be used. However, there will be six compulsory, **on-campus laboratories** (dates available in Canvas). All meetings, except for labs, will be conducted virtually over Zoom.

## Prerequisites

This course requires **CSCI2421 – Data Structures & Program Design.**
*Each student must sign the Prerequisites Agreement Form and take the non-graded prerequisite assessment to receive credit for any assessment. If this is not completed by Friday 5pm on the first week, the student will be administratively dropped from the course.*

**Prerequisites Agreement form and assessment will be available on Canvas**

## Learning Outcomes

**Expected Knowledge at the Start of the Course:**

Students are expected to understand the basic concepts of Object-Oriented Programming and the basic concepts behind Data Structures and Algorithms. Students are expected to understand the basic concepts of Program Construction.

**Expected Knowledge Gained at the end of the Course:**

Students are expected to understand

- Java Programming Concepts and Constructs,
- Python Programming Concepts and Constructs,
- multithreading and concurrency issues on parallel programs and
- the design of distributed multi-platform applications.

**Student Outcomes**

This course will address the following criteria:

- Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
- Apply computer science theory and software development fundamentals to produce computing-based solutions.

## Textbooks

We will use two textbooks for this course are:

- **How to program Java**. Early Objects. By P. Deitel and H. Deitel. Pearson. *ISBN 13*: 978-0-13-474335-6. (**Required**)
- **Intro to Python for Computer Science and Data Science.** 1st Edition. By P. Deitel and H. Deitel. Pearson. *ISBN 13*: 978-0-13-540467-6. (**Required**)

Additional readings, for deepening knowledge, may be announced during the course.

## Required Tools

- Computer with a minimum of 8GB of RAM (16G recommended).
- Microphone and Web-Camera for video-conferencing. A smartphone or a tablet can be used for videoconferencing.
- Internet connection (high speed recommended)
- VPN Access. See OIT website for setting your VPN.

## Topics

- Introduction Java, Classes and Objects, Inheritance and Polymorphism.
- Introduction to Python, Classes, Libraries.
- Exception Handling.
- Data Structures and Algorithms: Collections, Lists and Dictionaries.
- **Multi-threading, Concurrency and Networking**.

## Course Schedule:

**Schedule** Fall 2021

| Week | Date | Topic | Readings | Lab - On Campus | Homework Due | Assessment |
|---|---|---|---|---|---|---|
| Week 01 | Aug, 23 2021 → Aug, 29 2021 | Course Intro & OO Review | Add. Material | | **Hwk #1:** Introduction & Env. | Pre-Requisite is Due |
| Week 02 | Aug, 30 2021 → Sep, 05 2021 | Intro To Java | Java Ch. 1 to 5 | | | |
| Week 03 | Sep, 06 2021 → Sep, 12 2021 | Arrays & Exceptions. Classes deeper look | Java Ch. 6 to 8 | | **Hwk #2:** Java Basis | |
| Week 04 | Sep, 13 2021 → Sep, 19 2021 | Inheritance and Polymorphism, Abstract Classes and Interfaces | Java Ch. 9 & 10 | **Lab #1:** Arrays and Exceptions | | |
| Week 05 | Sep, 20 2021 → Sep, 26 2021 | Generic Classes and Collections | Java Ch. 16 & 20 | | **Hwk #3:** Polymorphism | Quiz 1 |
| Week 06 | Sep, 27 2021 → Oct, 03 2021 | Java FX | Java Ch. 13 | | | |
| Week 07 | Oct, 04 2021 → Oct, 10 2021 | Networking | Java Ch. 28 | **Lab #2:** JavaFx & Collections | **Hwk #4:** Collections | |
| Week 08 | Oct, 11 2021 → Oct, 17 2021 | Multi-threading | Java Ch. 23 | | | Due: Project Proposal |
| Week 09 | Oct, 18 2021 → Oct, 24 2021 | Introduction to Python & Functions | Python Ch. 1 to 4 | **Lab #3:** Network. & mThread | | Quiz 2 |
| Week 10 | Oct, 25 2021 → Oct, 31 2021 | Lists, Tuples, Dictionaries and Sets | Python Ch. 5 & 6 | | **Hwk #5:** Network. & mThread | |
| Week 11 | Nov, 01 2021 → Nov, 07 2021 | Strings, Files and Exceptions | Python Ch. 8 & 9 | | | *Project progress report* |
| Week 12 | Nov, 08 2021 → Nov, 14 2021 | Object Oriented and Python | Python Ch. 10 | **Lab #4:** Data Structures | **Hwk #6:** Python Basis | |
| Week 13 | Nov, 15 2021 → Nov, 21 2021 | Networking and Multiprocessing | Add. Material | | | Quiz 3 |
| Week 14 | Nov, 22 2021 → Nov, 28 2021 | Array-Oriented Programming - Numpy & Plots | Python Ch. 7 | **Lab #5:** Network. & mThread | **Hwk #7:** Data Structures | *Project progress report* |
| Week 15 | Nov, 29 2021 → Dec, 05 2021 | *Fall Break* | | | | |
| Week 16 | Dec, 06 2021 → Dec, 12 2021 | Database-Focused Programming | Java Ch. 24 | **Lab #6:** Numpy & DB | **Hwk #8:** Arrays & DB | Quiz 4. Due: Project |
| Week 17 Finals | Dec, 13 2021 → Dec, 19 2021 | Project Presentations | | | | |

The previous schedule is tentative, and it may change. Please, check the current one on Canvas.

## Assessments

There will be four types of assessments in this course. Two of them to be completed individually, and two of them to be working with a team.

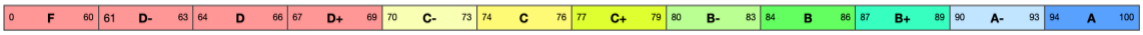
- **Homework**: homework is short to mid-size programming tasks that will assess the student's application of the materials presented in the previous weeks. **Homework will be automatically graded**. The student will have access to a platform to submit his/her homework submission, and this system will test the submission and inform of errors, if any, and the number of tests passed. The homework final grade will be a ratio between passes and total number of tests. Each homework may be submitted many times, to improve the performance. ***It is recommended to start homework asap****, so issues can be discussed with TA/Instructors.* **Homework is to be completed individually.** *The lowest graded homework will be dropped (excepting homework #1).*
- **Quizzes**: quizzes are usually a set of multi-choice questions that will assess the student's understanding of the materials presented in the previous weeks. **Quizzes are to be completed individually.**
- **Laboratories**: there will be six, on-campus, labs that will develop further the understanding of the materials. The week before the lab, an announcement on canvas will be sent to help students prepare for the labs. Labs are designed to be completed in pairs, though students may opt to complete the lab individually. However, assigned tasks will be the same whether completed alone or in teams.
- **Team Project**: students in teams of 2-3 teammates, will present a project proposal following a provided guideline. Then, the team will proceed to implement the project in a 7-week period. There will two short progress report to be completed and a final project presentation. Team project cannot be completed individually.

## Grading Policy

- Assignments will be **submitted through Canvas** (unless noted) on the dates and times posted.
- Lab attendance is mandatory to receive grade for the lab.
- In general, there will be no make-up homework or quizzes, though extraordinary circumstances should be discussed with the instructor previous the assessment is due.
- Late homework will not be graded, although special circumstances may be considered if emailed the instructor before the homework is due.
- The Final Grade will be distributed among the assessments following the table below.

| Assessment Group | Grade |
|---|---|
| Homework | 25% |
| Laboratories | 30% |
| Team Project | 35% |
| Quizzes | 10% |

- Final Letter Grade will be converted using the following scale:




## Communications

All announcements will be posted on Canvas. Make sure you have configured the notifications not to miss those. In addition, keep track of what is going on in the Slack/Discord chat, in this channel you may found useful hints for assignments.

## Collaboration and Cheating

I encourage you to review material and discuss ideas together for projects and other assignments, and to work on problems you encounter. It is a characteristic of computing that discussions often help to clarify problems and resolve difficulties — feel free to take advantage of this to improve your understanding of the material, and to complete projects, but **make sure you then create your own work**. It's important that you go through the program design, coding, and debugging processes yourself, or you will not be developing your own programming skills and understanding. "Working together" does not mean that one student does most of the work and other students put their names on it!

If you have any questions about what this means, please see me.

We reserve the right to use automated similarity metrics to detect plagiarism in this course.

**All students must create their work on their own!**

Any instances of cheating will result in a zero for the assignment, a grade of zero (an "F") in the course, or sanctions determined by the college (including suspension and expulsion).

Please, do not share solutions (whole or partial) on public channels like Slack/Discord.

All students must follow the College of Engineering, Design and Computing - Student Honor Code.