CSCI 3920: Advanced Programming using Java and Python
Course Syllabus
Fall, 2020

General Information

Instructor: Javier Pastorino
Email: javier.pastorino@ucdenver.edu
Phone: 303-315-1408
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 using 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 and 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 hosted on a remote asynchronous setting via pre-recorded lecture videos and online collaboration tools such as Canvas, discussion forums and Slack chat. However, there will be six compulsory synchronous laboratories (dates set in Canvas). These labs will be hold on Zoom. All meetings will be conducted virtually using Zoom as well.

Prerequisites

This course requires **CSCI2421 – Data Structures & Program Design.**
Each student must sign the Prerequisites Agreement Form to receive any credit for any assignment or exam. If this form is not signed by the first week, the student will be administratively dropped from the course.
Prerequisites Agreement form: Sign Here

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

The 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**)

Other books and readings, for deepening knowledge will be announced during the course.

Required Tools

- Computer (Windows, Mac or Linux) with a minimum of 8GB of RAM (16G recommended).
- Microphone and Web-Camera for video-conferencing. Can be omitted if a smartphone or a tablet is used for that purpose.

- 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.
- Lambdas and anonymous functions.

Course Schedule:

Week	Date	Topic	Readings	Lab	Homework Due	Assessment
Week 01	Aug 17, 2020 → Aug 23, 2020	Course Intro & OO Review	Add. Material		**Hwk #1:** Introduction & Env.	Pre-Requisite is Due
Week 02	Aug 24, 2020 → Aug 30, 2020	Intro To Java	Java Ch. 1 to 5			
Week 03	Aug 31, 2020 → Sep 6, 2020	Arrays & Exceptions. Classes deeper look	Java Ch. 6 to 8		**Hwk #2:** Java Basis	
Week 04	Sep 7, 2020 → Sep 13, 2020	Inheritance and Polymorphism, Abstract Classes and Interfaces	Java Ch. 9 & 10	**Lab #1**: Arrays and Exceptions		
Week 05	Sep 14, 2020 → Sep 20, 2020	Generic Classes and Collections	Java Ch. 16 & 20		**Hwk #3:** Polymorphism	
Week 06	Sep 21, 2020 → Sep 27, 2020	Java FX	Java Ch. 13			
Week 07	Sep 28, 2020 → Oct 4, 2020	Networking	Java Ch. 28	**Lab #2:** JavaFx & Collections	**Hwk #4:** Collections	PA#1 Release
Week 08	Oct 5, 2020 → Oct 11, 2020	Multi-threading	Java Ch. 23			
Week 09	Oct 12, 2020 → Oct 18, 2020	Introduction to Python & Functions	Python Ch. 1 to 4	**Lab #3**: Network. & mThread	**Hwk #5:** Network. & mThread	
Week 10	Oct 19, 2020 → Oct 25, 2020	Lists, Tuples, Dictionaries and Sets	Python Ch. 5 & 6			
Week 11	Oct 26, 2020 → Nov 1, 2020	Strings, Files and Exceptions	Python Ch. 8 & 9	**Lab #4**: Data Structures	**Hwk #6:** Python Basis	PA #1 is Due
Week 12	Nov 2, 2020 → Nov 8, 2020	Object Oriented and Python	Python Ch. 10			PA #2 Release
Week 13	Nov 9, 2020 → Nov 15, 2020	Networking and Multiprocessing	Add. Material		**Hwk #7:** Data Structures	
Week 14	Nov 16, 2020 → Nov 22, 2020	Array-Oriented Programming - Numpy & Plots	Python Ch. 7	**Lab #5**: Network. & mThread		
Week 15	Nov 23, 2020 → Nov 29, 2020	Fall Break	Fall Break	Fall Break	Fall Break	Fall Break
Week 16	Nov 30, 2020 → Dec 6, 2020	Database-Focused Programming	Java Ch. 24	**Lab #6**: Numpy & DB		
Week 17 Finals	Dec 7, 2020 → Dec 11, 2020	No Final.			**Hwk #8:** Arrays & DB	Due: PA #2

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

Grading Policy

- Assignments will be **submitted through Canvas** on the dates and times posted.
- Attending Labs meeting is mandatory to receive grade for the lab.
- Late homework will not be graded, although special circumstances may be considered if emailed the instructor before the homework is due.

Assessment Group	Grade
Homework	25%
Laboratories	30%
Programming Assignments	35%
Class Participation	10%

Notes:

Class participation relates to all activities that referred to the student's participation in the course such as participating in the discussions and collaborating with his/her peers.

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 the majority 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 rights to use automated similarity metrics in order to detect plagiarism in this course.

All students must create their own 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).

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