Course Number: IT114
Course Title: Advanced Programming for Information Technology
Section: 001, 003, 005
Semester: Fall 2023
Date & Time:

001 - Tues/Thurs 1pm-2:20pm GITC 3600
003 - Mon/Wed 11:30am-12:50pm ME 221
005 - Mon/Wed 1pm - 2:20pm TIER 107

Modality: Face to Face
Credits: 3
Office Hours: *Mon/Thurs 2:50 - 3:50 pm (CKB Main Floor Study Lounge, no appointment necessary)*

General availability via Discord via a provided communication channel

Course Catalog:
Prerequisites: CS 113 or CS 115.
Problem solving techniques and program design knowledge are expanded with an eye toward IT-related applications. Various kinds of data structures are introduced, including classic containers such as lists, stacks, queues, and trees. Sorting and searching techniques are examined. The fundamentals of client/server programming and the use of sockets are covered. Recursion and its various applications are studied. The built-in class library features of an object-oriented programming language are exploited throughout.

Instructor:
Matt Toegel (matthew.toegel@njit.edu)

Attending Class:

Synchronous:

Class will be held in the rooms and times given per your schedule from the registrar.
Mostly, I'll be sharing my screen with everyone and going over the topics either via the classroom projector or a screen-sharing service. There will commonly be time in class to practice the topic for that day and/or get a headstart on homework.

We'll be using Respondus for exams and everyone should ensure the software runs on at least 1 device (anticipate webcams will be required even in the classroom).

Asynchronous:

Class material will be available each Monday and is expected to be reviewed that week. Assignments may be due at the end of the week or the end of the next week. You'll also have participation assignments each week.

We'll be using Respondus for exams and everyone should ensure the software runs on at least 1 device (webcams will be required).

Both:

It’s highly encouraged to ask questions and express any doubts/concerns throughout the course. I want to give everyone the opportunity to raise any concerns or ask any questions to make sure they’re on track for the semester.

Make sure to always keep in communication with me if there are any concerns about the class or anything related, this can be done via Discord (preferred), email, Canvas Inbox, etc.

Academic Integrity:

The work done is expected to be your own, any group work should clearly distinguish ownership of tasks. Use of snippets/material from others should be kept to a minimum and the source should be accredited where applicable.
That being said, please also note the below:

Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the academic code of integrity policy that is found at:
http://www5.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf.

Please note that it is my professional obligation and responsibility to report any academic misconduct to the Dean of Students Office. **Any student found in violation of the code by cheating, plagiarizing or using any online software inappropriately will result in disciplinary action. This may include a failing grade of F, and/or suspension or dismissal from the university.** If you have any questions about the code of Academic Integrity, please contact the Dean of Students Office at dos@njit.edu” Any violations of the NJIT Honor Code will be brought to the attention of the Dean of Students.

Overview:

This course will utilize the Java programming language to demonstrate/expand on data structures, client/server programming, and various other problem solving skills such as recursion. You’ll be able to utilize Java libraries, carry out inter-process communication via sockets, and develop a recursive solution for some common problems. There may be opportunities to dive deeper into certain topics or include other topics per general interest of the class. After the basics, the class will focus on a solo project that students will pick from a list that will demonstrate a multiple client socket server.

Outcome:

At the end of the course, each student will have the skills and knowledge to build and deploy Java applications. Experience will include use of version control via git/github which will be used to record the progress of an individual project (per student and picked from a preset selection). The project should make for a good portfolio piece and/or a significant stepping stone for future classes.

Illustrative Schedule

The schedule is a guideline and is subject to change to fit the particular instance of the class. All topics in general are planned to be covered. Some may have more focus than others and per class interest other topics may be included. Note: Some modules may span more than one week, but in general they'll be about 1 week in length and extra time later in the semester will go towards Project Topics and Questions.

Module 1: Overview of Class / Environment Setup / GIT Intro / VS Code Intro
Module 2: Review Java Basic Data Types / Arrays / Lists / Queues
Module 3: File I/O / Objects and OOP Concepts
Module 4: Trees / Recursion / Project Discussion
Module 5: Intro Client/Server Sockets / Project Choices
Module 6: Continued Client/Server Topics / Project Milestone 1 / File I/O Integration
Module 7: Java UI Intro / Project Milestone 2
Module 8: Expand on Project Related Topics, Project Milestone 3
Module 9: Project Topics / Deployment
Module 10: Final Demo / Final Deliverable

Assignments:

Each week there will be coding samples related to the current week's topics. There will also be supplemental online resources as well as recordings. There will be a semester-long project that each student will incrementally develop during the semester as new topics are learned (a set of requirements/objectives will be given via a Proposal document). The project will be based on an agreed-upon proposal and the material covered in class. During the semester there will be milestone deliverables for groups of features from the project. The milestones will cover the gist of the features; there commonly is some time between the last milestone and the final demo/deliverable where the remaining features can be implemented and/or cleaned up. Projects may have a chance for extra credit based on surpassing the minimal requirements of the proposal at the discretion of the instructor.

Visual Studio Code will be the IDE of choice with the Java extensions installed for ease of development/debugging. Typically, we'll be compiling and running our code from the CLI as the code complexity increases.

You're free to use any other IDE as long as it supports the extensions mentioned during class.

Some assignments will be small direct topics while others may combine multiple learnings.

Grading:

Exams / Tests will be graded out of 100 points.
Quizzes will be graded out of 10 and will typically have only one attempt per quiz
Projects / Assignments will be graded out of 10. Some items may have opportunities for extra credit which will be determined per assignment and at the discretion of the instructor.
All points will be converted to a final percentage and letter grade at the end of the semester. Canvas will already have the weightings applied.

Grading Breakdown:

- Midterm: 20%
- Quizzes: 15%
- Participation/Attendance: 5%
- Assignments: 10%
- Milestones: 25%
- Final Project Deliverable (25%):
  - Completed Project (Last Milestone and remaining features)
  - Final Demo

Extra credit may be given for exceptional programming projects.

Grading Scale:

A	100 % to 89.5%
B+	< 89.5 % to 84.5%
B	< 84.5 % to 79.5%
C+	< 79.5 % to 74.5%
C	< 74.5 % to 69.5%
D+	< 69.5 % to 64.5%
D	< 64.5 % to 59.5%
F	< 59.5 % to 0.0%

Materials/Technologies:

VS Code (or preferred IDE)
Google Compute Engine, AWS EC2, etc
Java SDK (latest)
Online Resources (provided)
Canvas
GitHub/Git

Late Policy:

You'll have a two business day grace period before lateness is applied (Canvas will automatically apply the deduction but during the grading process up to two days of lost points will be returned)

All deliverables will be eligible for a 5% penalty per day late. Typically this will be controlled by Canvas.

Late assignments will automatically be marked as 0 by Canvas and will be pending until they've been graded.

Missed Exams/Quizzes will result in a 0.

If you are going to miss a class/material and cannot hand in an assignment, it's your responsibility to let me know as soon as possible so the situation can be handled.

There also will be no make-up exams (except, at the discretion of the instructor in the case of a documented medical or family emergency from the Dean of Students).

For any emergency please reach out to the Dean of Students so they can send out an official notice.

Syllabus is subject to change, attend class to stay current.