Course Syllabus: IT114 - Advanced Programming for Information Technology

Matt Toegel – matthew.toegel@njit.edu

Table of Contents

1. General Information

Course Number:	IT114
Course Title:	Advanced Programming for Information Technology
Section(s):	003, 005
Semester:	Fall 2024

Date & Time:	**003:** Monday/Wednesday 11:30 am - 12:50 pm **005:** Monday/Wednesday 1:00 pm - 2:20 pm
Modality:	Face-to-Face
Credits:	3
Office Hours:	**CKB Public Area/Lounge Main Floor:** Monday/Wednesday 10 am - 11:20 am General availability via Discord via a provided communication channel

2. 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.

3. Course Catalog

Prerequisites: CS 113 (https://catalog.njit.edu/search/?P=CS%20113) or CS 115 (https://catalog.njit.edu/search/?P=CS%20115).

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.

4. Instructor

Matt Toegel
Email: matthew.toegel@njit.edu
Discord: MattToegel
Github: MattToegel

5. Prerequisites

This course is an Advanced topics course based on the Java programming language, although some basics will be reviewed in the beginning it's anticipated that students have a reasonable computing background (in addition to the Course Catalog Prerequisites). Students should be familiar with file system structure, running programs from the command line, installing programs, editing text files, etc.

If you are not well versed in these subjects, here are some resources that may help you get up to speed:

GCFGlobal - Basic Computer Skills (https://edu.gcfglobal.org/en/basic-computer-skills/)
Debugging: The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems (https://www.amazon.com/Debugging-Indispensable-Software-Hardware-Problems/dp/0814474578)
codeacademy - Operating Systems: Filesystems (https://www.codecademy.com/learn/operating-systems-filesystems)

Visual Studio Code - Getting started with Visual Studio Code (https://code.visualstudio.com/docs/introvideos/basics)
simplilearn - What is Client-Server Architecture? Everything You Should Know
(https://www.simplilearn.com/what-is-client-server-architecture-article)

6. Attending Class

6.1. 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).

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.

7. Learning Outcomes

1. Students can use Java as a general-purpose programming language
    a. Students can install and run Java
    b. Students can design an algorithm to solve a problem
    c. Students can run and write complex Java programs
    d. Students can follow standard conventions for software development
2. Students can utilize Java packages and OOP concepts
    a. Students can utilize built-in Java packages effectively
    b. Students can break down their code into modular components
    c. Students can structure projects for complex scenarios
3. Students can utilize Lists, Queues, Stacks, Trees, and recursive concepts
    a. Students can use these in small programs to accomplish challenges
    b. Students can use these in their semester project within the complexity of Client/Server architecture and UI
4. Students can utilize Sockets to create client/server architecture applications
    a. Students can use Java Sockets to connect to a ServerSocket
    b. Students can use ServerSocket to receive Java Socket connections
    c. Students can utilize the bi-directional channel for communication of complex data
5. Students can utilize Java UI packages
    a. Students can construct User Interfaces
    b. Students can build complex views with cross-communication of data
6. Students can problem solve and debug
    a. Students can utilize the Java Debugger and other debugging techniques

b. Students can formulate their own code to solve complex problems

c. Students can analyze, interpret, and integrate course-provided code

7. Students can utilize version control

a. Students can use git commands via the terminal/command line

b. Students can use GitHub for managing histories of their work

8. Students can gain experience with a milestone approach of building a complex project

a. Students can incrementally develop a large project in a sprint-like structure (based on provided proposals)

b. Students can extend existing code to add new features

c. Students can add new classes to solve requirements

d. Students can integrate most course concepts (OOP, Data Types, Sockets, UI, Flow Control, etc)

8. 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.

Each Milestone will generally have 2 weeks to work on the specific requirements.

Module	Topics
Module 1:	• Overview/Introduction of Course • Environment Setup (VS Code/JDK) • Git/GitHub Intro
Module 2:	• Review Java Basic Data Types • Flow Control • Loops and Iterations • Arrays, ArrayLists, Lists, Queues, Stacks
Module 3:	• User Input • OOP Concepts (Modifiers) • File I/O • Recursion/Trees
Module 4:	• Sockets Intro (Parts 1 - 3) ◦ Client/Server concepts • OOP Concepts (Encapsulation, Packages, Inheritance) • Project Choices • Midterm Prep

Module	Topics
Module 5:	• Sockets Intermediate (Parts 4 - 5) • **[Midterm]** • **[Project Milestone 1 Introduced]**
Module 6:	• Preparing for Milestone 2 • Build/Run scripts • Milestone 2 Concepts ◦ integrating concepts from Modules 1 - 3 • **[Project Milestone 2 Introduced]**
Module 7:	• Continued Milestone 2 concepts/topics • Project Flow and Class Responsibility
Module 8:	• Milestone 2 wrap up before due date • Java UI Intro (Swing/AWT) • Milestone 3 Prep • **[Project Milestone 3 Introduced]**
Module 9:	• Integration of Java UI to existing Milestone 2 • Milestone 3 Concepts (Integrating project specific UI components and features)
Module 10:	• Milestone 4 Concepts • Project Server Deployment Concepts • **[Project Milestone 4 Introduced]**
End of Semester:	• Final Milestone Due (deliverable) • Final Demo of Project Due (presentation)

9. Assignments

Each week there will be coding samples related to the current week's topics. Additionally, there will be supplemental online resources as well as recordings available to support your learning.
Assignments generally are graded out of 10 points unless otherwise noted.

9.1. Semester-Long Project

There will be a semester-long project that each student will incrementally develop as new topics are learned. A set of requirements/objectives will be given via a Proposal document at the start of the semester.

The project will be based on an agreed-upon proposal and will cover the material discussed in class. During the semester, there will be milestone deliverables for groups of features from the project. These 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.
Milestones are graded out of 10 points.
Students are expected to utilize the course material and structure to work on the projects as this aids/eases learning/debugging and enhances perspective for anyone already familiar with a particular design/structure.

9.2. Quizzes

There will be weekly online quizzes that'll test the current module's topics. These will be taken via Canvas online by the designated due date (typically the following Monday night of the particular module). Each quiz requires a passphrase, generally has one attempt unless otherwise noted, and will generally be capped at 10 minutes. Scoring is out of 10 points. These are expected to be closed notes and taken solo. Part of the goal is to highlight the topics that require extra review/study as they'll come up again in the semester.

9.3. Exams

All exams will use Respondus, so be sure to bring a compatible device with you on the day of the assessment. The midterm will take place during a regular class period and will cover the material from modules one to four. Exams will be closed book and must be taken in the classroom if the class is meeting face-to-face.
Any exams will be graded out of 100 points.

10. Grading

10.1. Breakdown

Midterm: 20%
Quizzes: 15%
Participation/Attendance: 5%
Assignments: 10%
Milestones (1-3): 25%
Final Project Deliverable: 25% (Includes last milestone and remaining features)
Final Demo: Included in final project deliverable

All points will be converted to a final percentage and letter grade at the end of the semester. Canvas will already have the weights applied.

10.2. Extra credit

Extra credit may be given for exceptional programming projects at the discretion of the instructor.

10.3. Grading Scale

Grade	Percentage Range
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%

Grade	Percentage Range
D+	<69.5% to 64.5%
D	<64.5% to 59.5%
F	Below 59.5%

11. Materials/Technologies

1. To better match industry trends and standards, this class will utilize online resources in lieu of a traditional textbook. Resources will be provided as the course progresses, but some primary resources are listed here:
    a. w3schools - Java Tutorial (https://www.w3schools.com/java/)
    b. sololearn - Java Developer (https://www.sololearn.com/en/learn/languages/java) (additionally recommended)
2. This class utilizes Canvas learning management system. You can find assignments, assessments, and learning resources there.
3. This class utilizes a platform for assignment evidence gathering and submission (provides clear objectives and generates a file to upload to Canvas for review)
    a. Students will access the worksheets through Canvas to fill in the required deliverables and the final submission will be uploaded to Canvas
    b. The worksheets aim to maintain clarity of requirements for the student and for objective and fair grading
4. This class will utilize the Respondus proctoring system for any/all exams.
5. Students are required to have a device that meets the YWCC minimum specifications (https://ist.njit.edu/student-computers-recommended-specs). They will need administrative acess to this device to install software. The device should be functional (charged, working) and brought to class.
6. Students will utilize an installed IDE, (VS Code (https://code.visualstudio.com/Download) is recommended and will be used by the instructor)
    a. Students may use another IDE at their discretion as long as it supports the necessary complex structure that'll be utilized in addition to any required extensions
7. This class has a GitHub page (https://github.com/MattToegel/IT114) where you can find out how the course materials are built and have changed overtime.
    a. There are likely many branches included, focus should remain on the branches with the prefix ending in the most recent year's last two-digits.
    b. Will aim to keep the main README.md updated with the most recent/relevant branches.
8. Students will install and utilize the latest JDK 22 (https://www.oracle.com/java/technologies/downloads/#java22)
9. Students will utilize git and github for tracking development progress
    a. The first module's lessons will be dedicated to the setup of these resources and any naming conventions to follow
    b. git for windows (https://git-scm.com/download/win)
    c. git for mac (https://git-scm.com/download/mac) (Homebrew is recommended)
    d. git for linux (https://git-scm.com/download/linux)

e. Github Education (https://education.github.com/discount_requests/application) (the education package is recommended, but a regular free account is sufficient)

12. Policies

12.1. 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.

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: academic integrity code (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.

12.2. Requesting Accomodations

If you are in need of accommodations due to a disability please contact the Office of Accessibility Resources & Services (OARS) (https://www.njit.edu/studentsuccess/accessibility), Fenster Hall Room 260 to discuss your specific needs. A Letter of Accommodation Eligibility from the OARS authorizing your accommodations will be required.

12.3. Resources for NJIT Students

NJIT Services for Students (https://docs.google.com/document/d/1xGO2qcVEF1tsOgZn-_W1LjSOKn_jhEVs9IWI_6jeuPs/edit?usp=sharing), including Technical Support

12.4. Class Etiquette

Students who are the most successful attend and participate in class. If you have questions, please ask them. This makes the class more dynamic and interesting for everyone.

12.5. Proctoring

NJIT policy requires that all midterm and final exams must be proctored, regardless of delivery mode, in order to increase academic integrity. Note that this does not apply to essay or authentic based assessments. Effective beginning Fall semester 2019, students registered for a fully online course section (e.g., online or Hyflex mode) must be given the option to take their exam in a completely online format, with appropriate proctoring.

Exams will be given in-person using Respondus. Be sure to bring your charged laptop and charger on the day of exams.

12.6. Late Policy

All deliverables will be eligible for a 5% penalty per day late, generally Canvas automatically has this set and will automatically apply it as grades are entered. Late assignments will automatically be marked by Canvas as a 0, but will be updated once grades are entered for the particular item. Exception: There will be a manual two-business-day grace period applied to reduce the penalty if incurred.

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** (https://www.njit.edu/dos/student-absence-verification) so they can send out an official notice.

13. Closing Notes

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

Last updated 2024-09-09 15:10:41 UTC