

CSCI 392 - The Java Programming Language Fall 2019

Department of Computer Science and Quantitative Methods
College of Business Administration
Winthrop University

Instructor: Stephen Dannelly, PhD
Office: 315 Thurmond
Email: dannellys@winthrop.edu
Home Page: faculty.winthrop.edu/dannellys/csci392
Office Hours: Mondays and Wednesdays 1:00-4:00
see homepage for weekly schedule

Credit Hours : this is a 1 credit course

Catalog Description

An introduction to the Java programming language.

Course Objectives

After successfully completing this course, the student will understand and be able to demonstrate:

- basic Java programming principles, such as file IO and exception handling
- object creation and inheritance in Java
- use of advanced Java packages
 - image processing
 - client / server communications

Prerequisites

- CSCI 271, or permission of the instructor.
- All CSCI courses numbered above 299 have a prerequisite of junior status, an overall GPA of at least 2.00 and a grade of C- or better in HMXP 102.

Course Grades

Final grades for this course will be based on a 10-point scale. In other words,
90.00 to 100 is an A,
80.00 to 89.99 is a B,
etc...

This course heavily emphasizes the practical application and implementation of advanced programming concepts. Hence the course grade is heavily based on programming assignments.

Programming Assignments: 80%
Final Exam: 20%

Some assignments are more complex than others, and so not all assignments will count the same number of points. There will be approximately ten programming assignments.

Work submitted late will lose 10%, plus 10% for every 12-hours submitted late. For example, if an assignment is due Thursday at Noon and is submitted at 12:10pm, it loses 10%. Work submitted at 12:30am on Friday would lose 20%.

Grades for programming work will be assigned using the following rubric:

| | Excellent | Good | Fair | Unsatisfactory |
|------------------------------|--|---|---|---|
| Functionality 40% | The program works and meets or exceeds all specifications. | The program works and produces the correct results and displays them correctly. It also meets most of the other specifications. | The program produces correct results but does not display them correctly. | The program performs tasks that it was not supposed to perform. |
| Efficiency 20% | The code is extremely efficient without sacrificing readability and understanding. | The code is fairly efficient without sacrificing readability and understanding. | The code is brute force and unnecessarily long. | The code is huge, inefficient, or appears to be patched together. |
| Robustness 20% | Program correctly handles all specified test cases and additional special cases. Program contains error checking code. | Program executes without errors and correctly handles most special cases. Thorough testing has been completed. | Program executes without errors for pre-specified test cases. | Program does not execute due to errors. No evidence of testing. |
| Documentation 20% | Clear and complete documentation. The purpose and constraints of every variable and subroutine are described. Comments for code segments make the code easy to follow. | The purpose of all variables is clearly explained. The purpose of each subroutine is described. | Most variables and subroutines are commented. Comments provide some assistance with understanding the code. | No comments are included or other documentation provided. |

Cheating

When working in a professional software development environment it is expected that you will frequently consult with your colleagues regarding problems you encounter. But in an educational environment, each student is expected to do all of his/her own work.

In freshmen courses, such as CSCI 207, you were encouraged to help each other debug code. That was because overcoming syntax was a problem common to all students and you could frequently learn from others' mistakes. In sophomore courses, such as 271, debugging help was not allowed while sharing design ideas was allowed. In this upper-division course, you are expected to be able to design the entire application by yourself and you are expected to be proficient at debugging your own code.

You are still allowed to discuss problems and strategies at a high level, but sharing code, pseudo-code, or algorithms is not acceptable. For example, you are allowed to tell a classmate, "I couldn't determine if my app was writing into my data file in the correct location, so I wrote a separate short app to simply display the entire data file." But showing a classmate your code to dump a data file to the screen, discussing the format of your data file, or helping debug his/her code, is considered cheating.

Tentative Schedule of Topics

| | |
|---------------------------|--|
| first half of semester | language constructs: selection, iteration, IO, classes, inheritance |
| third quarter of semester | project one: java's image processing library |
| final quarter of semester | project two: using java's client/server library to write a web crawler |

See course web site for day-to-day topics, lecture notes, example code, assignments, etc.

Final Exam - 8:00am, Monday, December 9, 2019

COLLEGE OF BUSINESS EXPECTATIONS REGARDING PROFESSIONALISM IN THE CLASSROOM

The College of Business Administration is a professional organization with a well-defined and widely disseminated mission of student development. Accordingly, each class represents a gathering of professionals and professionals-in-training. The instructor's job as a professional is to deliver quality instruction in each class, to start and end each class on time, to be responsive to student perspectives, issues and questions, and to treat each student respectfully. The student's job, as a professional-in-training is to be prepared for class, to be on time, to attend all classes, and to be respectful of others in the classroom.

In accordance with and pursuant to these roles the following guidelines were established to specify to students (both present and prospective) faculty expectations regarding their behaviors

1. **Students will attend all class meetings.** There are no automatically "excused" absences. In the event that you will be unable to attend a class session, you should inform your professor in advance as a matter of professional courtesy just as you would/should with an employer.
2. **Students will arrive in advance of the beginning of the class session.** Late arrivals are disruptive, inconsiderate and unprofessional. Professors may make arrangements for delinquents, but are not obliged to do so. Those not present at the beginning of the classroom period will be considered absent.
3. **Students will not converse among themselves during class except when instructed to do so.** When a student creates a disturbance in the classroom, instructors will either ask the student to desist immediately or speak to the student at the conclusion of class. Repeat offenders will be sanctioned.
4. **Students will not leave class before its conclusion.** Early departures are disruptive, inconsiderate and unprofessional. Professors may make arrangements under some circumstance, but are not obliged to do so. Those not present at the conclusion of the classroom session will be considered absent.
5. **Students will have procured textbook/materials prior to the first class.** Instruction will begin with the first class meeting and consume the remainder of the class period.

STUDENTS WITH DISABILITIES

Winthrop University is committed to providing access to education. If you have a condition which may adversely impact your ability to access academics and/or campus life, and you require specific accommodations to complete this course, contact the Office of Accessibility (OA) at 803-323-3290, or, accessibility@winthrop.edu. Please inform me as early as possible, once you have your official notice of accommodations from the Office of Accessibility.

SYLLABUS CHANGE POLICY

The grading and attendance policies for this course, as described above, will not change and are adhered to strictly. The schedule of class meetings may change due to unexpected events such as class cancellation due to weather.

ATTENDANCE POLICY

Because programming assignments and the final exam will be based on the content of lectures, not a textbook or the web lecture notes, it will very difficult for students to successfully complete the graded work without coming to class.

ASSESSMENT

No student performance metrics from this course are used to assess any program.