Programming with Objects and Data Structures

Course Description: Assumes a foundation in procedural programming as covered in CS 110. Provides the foundations of software development using Java and the data structures provided by Java. Problem solving using object-oriented programming techniques is emphasized. Topics include primitive and reference data types, variables, expressions, assignment, functions/methods, parameters, selection, iteration, recursion, exception handling, generic linear data structures, trees and maps, file types, file I/O, simple GUIs, programming to an interface, use of inheritance, Javadoc documentation, and introduction to Java streams and threads. Required laboratory provides supervised problem solving, programming using the command line as well as Eclipse or Netbeans development environments, code backup in a version control repository, debugging and JUnit testing techniques.

Prerequisites: All prerequisites must have a grade of C- or better.

- CS 110, CS Majors may request a waiver from the Undergraduate Director based on prior programming experience.
- Math 225.
- CS 120. (May be taken concurrently)

Credits /Contact Hours: 4 credits, Three 60-minute lectures and one 85-minute lab per week
Lecture: 2:20pm – 3:20pm (MWF)
Labs: Tuesday Afternoon at LNG 103

Webpage: http://www.cs.binghamton.edu/~tbartens/CS140_Fall_2019/

Teaching Staff: Professor: Tom Bartenstein. Teaching Assistants: See web page for details.

Textbook: *Big Java Early Objects*, 6th Edition by Cay Horstmann, John Wiley and Sons (2015)
Java API, Java Tutorial and other information available on-line (see class web page)

Effort Expected: As announced by the Faculty Senate, Binghamton University expects a minimum of 12.5 hours per week of student effort for a 4-credit course. Of these 12.5 hours, 3 are in class, including exams during class periods and 1 hour 25 minutes are in the Lab.

Lecture Notes: Lecture Notes for each lecture will be posted on the class web-site in PDF format before lectures. *Lecture notes do not substitute for class attendance*, since (i) they will not be complete and (ii) significant parts of lectures, including discussions and in-class exercises, may not come from the lecture notes.

Course Objectives: Upon completion of this course students will understand and be proficient in the use and application of:

- Programming in Java using both the command line and an IDE
- Declaration, types and assignment of primitive variables in Java.
- A variety number types and their range and precision.
- Control flow constructs: if statements, while loops, for loops, enhanced for loops
- Methods and their parameters, return values. Method calls and arguments. Lambda Expressions
- Arrays
- Simple recursive methods
- I/O for the console. Reading and writing binary and text files
- Interfaces, Classes and Objects, implementation of interfaces, subclasses and inheritance.
- Variables declared as reference types and the concept of the run-time type (dynamic type) of a variable
- Lists (especially ArrayLists), Maps, and Trees.
- Programming with Streams
- Javadoc comments
- Overloading and overriding of methods. Polymorphism (dynamic dispatching of method calls)
- Simple JUnit testing and basic code debugging
- Exceptions and exception handling

In addition, students will see an introduction to:

- The call stack and activation records
- Diagrammatic representation of the run-time structure of objects and the connecting references between them
- Sorting and searching of arrays
- Big-Oh notation
- Definition and use of enumerated types in Java
- Java timers for animation.
- Java Swing components and layout managers
- Event handling and user interaction using GUI interfaces supplied by the instructor.
- Java Modules
- Identification of classes and methods in the design of object oriented software.

Grading: Your grade will be based on:

In Class and Lab Tests	30%
Lab Results	15%
Assignments	15%
Project	10%
Quizzes / class and lab participation / attendance	10%
Final Exam	20%

You can find your current weighted numeric average on MyCourses, but the class and lab participation will not be posted on MyCourses, so the weighted average on MyCourses is not your complete grade. The mapping from a weighted average to a letter grade changes from semester to semester based on the difficulty of the tests, labs, and assignments as compared to previous semesters; and will not be published.

Attendance and Participation: Formal attendance may not be taken for each lecture or lab, but attendance is expected. There will be un-announced quizzes administered throughout the semester during lecture or lab periods. Unexcused absence from a quiz will result in a zero grade for that quiz. It will not be possible to make-up quizzes. If you cannot attend a lecture or lab, e-mail the professor *before* the lab or lecture, and make sure you check afterwards to ensure you know what you missed. Your participation grade will include of an average of the quiz grades, where the numerator is the sum of the quiz grades you received, and the denominator is the number of *unexcused* quizzes. Another component of the participation grade will be the professor's subjective evaluation of participation, including questions or comments in class and lab, and extra help given or received during office hours or via e-mail.

Assignments: Assignments are individual assignments. This means that all work submitted will have been done by you. You may (and are encouraged to) seek help from others, including the instructor, TAs, and classmates. Help includes assistance with:

- using the software tools needed to complete an assignment
- understanding the specifications or requirements of an assignment

Get TA or instructor help with:

- evaluating strategies for solving a problem
- debugging code that you have written
- interpreting compiler and run-time error messages

Obvious violations of the principles of academic honesty include, but are not limited to:

- submitting another person's work (in whole or in part) as your own
- submitting the same work (with or without minor changes such as changes of names and the order of code sections) as another student

Assignment instructions and due dates will be published on the class web page. Assignments are due no later than 11:59pm on the due date. Assignments submitted after this deadline will be treated as follows:

- submitted late, but within three days of the due date: 20% deducted from the grade
- After three days, no credit without approval from your instructor.

Lab Results: Most labs will require students to submit write-ups and/or code via MyCourses and gitHub. As in the homework, all submissions are individual assignments. This means that all work submitted will have been done by you. Lab submissions are designed to be completed during the lab period itself, and will be due by 11:59 PM Tuesday evening, the day of the lab.

Getting Help: Please utilize the instructors and TA's office hours for questions and discussion of course related material. Our job is to make you successful, and office hours are a great way to get help. E-mail the instructor or TA's if you need an appointment outside of office hours, or e-mail questions or discussions to the professor directly. See the class web page for tutoring options as well.

Academic Honesty Expectations: Please review the academic honesty document and make sure that you understand it! The link is at: http://www.binghamton.edu/watson/about/honesty-policy.pdf. Cheating and copying will NOT be tolerated. For instance, any code turned in will be compared to other students' submissions. If there is significant similarity, even if there are different variable names or comments, all such similar code will receive a zero grade.

Collaboration: Students are encouraged to help one another and to form study groups. In Computer Science, you can learn more from your peers than from your instructors and teaching assistants. As long as the help is appropriate, please be generous with your time and expertise when helping fellow students. Doing so is good for you and good for them. You are free to discuss assignments *in general terms* with one another. However, please do not show your work directly to other students. Do not store your work on publicly available repositories (such as public GIT repositories.) Each of you must write your own code, and you must write up all solutions individually. Students submitting solutions (including code) that are determined to be "too similar" are likely to be punished equally and harshly.

Academic Stress: If you are experiencing undue personal or academic stress at any time during the semester or need to talk with someone about a personal problem or situation, I encourage you to seek support as soon as possible. I am available to talk with you about stresses related to your work in my class. Additionally, I can assist you in reaching out to any one of a wide range of campus resources, including:

- Dean of Students Office: 607-777-2804
- Decker Student Health Services Center: 607-777-2221
- University Police: On campus emergency, 911
- University Counseling Center: 607-777-2772
- Interpersonal Violence Prevention: 607-777-3062
- Harpur Advising: 607-777-6305
- Office of International Student & Scholar Services:607-777-2510