CS – 140
Programming with Objects

Instructor: Tom Bartenstein

Course Web Page:
http://www.cs.binghamton.edu/~tbartens/CS140_Fall_2020/

CS-140 Programming with Objects

Course Description

Catalog Description

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,
- 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.

Course Goals - Proficient

- 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

Course Goals - Introduction

- 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.

Expectations

- Pre-Requisites
  - Assumes a foundation in procedural programming as covered in 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)
- 4 Credit Hour Course implies 12-12.5 hours per week
  - Lectures/Discussions/Meetings/Labs – 4.5 hours
  - Assigned readings
  - Assignments
  - Study for Tests and Exams

Textbook & Other Resources

- *Big Java Early Objects, 6th Edition* by Cay Horstmann, John Wiley and Sons (2015). Big Java Early Objects Web Resources
- Textbook is a parallel resource to the lectures
  - Two learning streams is better than one
- Cross references in slides using : Chap X.Y
- "Supplemental Material" on the class page for on-line references
  - Free textbooks, tutorials, etc.

Teaching Staff

- Prof: Tom Bartenstein (tbartens@binghamton.edu)
  - Office Hours: Monday and Wednesday, 3:30 - 5:50 PM via Zoom Office Hours or by email appointment
  - Office : EB Q06 (behind CS secretary) [Only if necessary]
  - e-mail
  - By appointment
  - See the Meet the Professor video
- Teaching Assistants
  - Paperwork in progress

CS-140 Course Mechanics

Lectures

- Lectures will be delivered via on-line videos
  - Available at the "Lectures" link from the class web page
  - Each video is between 5 and 20 minutes long
  - More videos will be added as the semester progresses
- A "due date" will be published
  - You are expected to view and understand the video by that date
- PDF of lecture notes will be posted on class web page
  - PDF will be updated if errors need to be corrected
- Discussion questions and tutorial references also published

Discussion Groups

- I will divide the class into three discussion groups
  - Discussion group M meets Monday 2:20 - 3:20 (Eastern Time)
  - Discussion group W meets Wednesday 2:20 – 3:20 (Eastern Time)
  - Discussion group F meets Friday 2:20 – 3:20 (Easter Time)
- Discussion groups will meet (synchronously) via Zoom:
  - High level of student participation expected
  - Attendance via Zoom Statistics
  - Review videos due this week
  - Answer student questions
  - Investigate lecture discussion questions
  - Discuss homework and lab requirements, issues, etc.
  - Occasional unannounced short quizzes

To ask or not to ask?

- Instructors are not psychic!
  - If you don't ask, I don't know you need help
  - If you don't ask, the other 5 students who have the same problem won't get help
- Please let me know if…
  - Your are lost
  - You don't understand something
  - You don't have the background
  - Class can be improved

- Ask for help early – don't wait until the test or exam!
- Give feedback! The more feedback – the better the course!

Labs

- All labs will be done by small lab groups of 2-3 students
- Lab groups will be assigned weekly and randomly from your lab section (B55 or B56)
- The lab will be held synchronously via Zoom
  - Each lab group will be in a Zoom breakout room
- CA and/or Professor will rotate between groups to answer questions, provide assistance, nudge in the right direction
- Each group will submit a single solution
  - All members of the group get the same grade
  - Zero for students not present without Prof. permission
- 10% penalty if lab is submitted after the assigned lab period

Homework Assignments

- Mostly short programming assignments
  - Occasional short answer and test style questions
- Posted on class web page and announced in discussion groups and/or on myCourses announcements
- Not quite weekly (depending on how much time I have)
- At the end of the semester, will be stopped so you can work on the final project

Final Project

- Large programming assignment in lieu of homework at the end of the semester
- Lab time will be devoted to the final project
- Teams of 1, 2, or 3 students (you choose your partner(s))
- Most of the code will be provided to you
- Your team will be responsible to complete the code
  - Typically – three phases, each submitted and graded separately
- Questions on your solution will occur in the oral part of the exam
  - To verify that all members of the team understand all aspects.

Accounts Used in this Class

- BU Email account (<username>@binghamton.edu)
- Access to myCourses for assignments and grades
- Zoom (Binghamton University Account)
- Piazza

Office Hours / Assistance

- Zoom link / time posted on class web page
- Use Zoom "waiting room" to maintain privacy
  - If you want to meet as a group, that's fine… just let me know
- Ask questions after lectures (recording will be turned off.)
- Class bulletin board - Piazza
- Outside of posted office hours, email for appointment
- email questions directly (attach code for debug help)
- Make Google your friend. Can’t beat the response time!

Tests and Exams

- Two 1 hour open book on-line mid-term tests
  - Scheduled to replace one Discussion Group session
  - Test 1 Covers Methods, Fields, up to Type Conversion (1st third)
  - Test 2 Covers Object Oriented Concepts up to ? (GUI?) (2nd third)
- One 2 hour open book on-line final exam
  - Covers entire semester
- One final oral exam
  - Ten minute interview by the instructor to ensure mastery
  - Scheduled during lecture time in final week (after Thanksgiving)

CS-140 Course Policies

Academic Honesty for On-Line Courses

- Motivation: Increase my GPA
  - Everybody else is cheating, if I don't cheat, I'm just being stupid
- Opportunity: On-line tests (even open book)
  - It's almost impossible for anybody to catch me cheating
- Means: On-line cheating web sites
  - Near immediate answers to any question
  - e-communicate with other students

Flaw in Cheating Logic

- The goal of getting a degree is NOT to get a high GPA!
  - The goal is to master the material
- A high GPA helps if GPA is a valid indicator of your mastery of the material
- If everybody cheats, everybody gets a high GPA, whether they have mastered the material or not
- People and Institutions that used GPA to distinguish between candidates are finding that GPA is not as useful
- More attention given to interviews, recommendations, etc.

My Teaching Philosophy

- My primary goal is to make sure as many students gain as much knowledge as possible!
  - Cheating erodes my capability to achieve this goal
- I use grades to motivate learning
  - **If you understand the material**, you will get a good grade!
  - Studying for and taking a test reinforces learning
  - Cheating de-motivates learning
- My secondary goal is to ensure that grades correctly reflect mastery of the material
  - Cheating erodes my capability to achieve this goal!

Academic Honesty

- For Yourself!
  - College education is not cheap - Don't waste it!
  - You will eventually get caught
- For your peers
  - One person cheating hurts all other students
  - If you cheat, your friends will get a lower grade
- For me
  - Cheating wastes time and effort
  - Cheaters hurt my reputation
- For the institution
  - Our reputation depends on you!

Academic Honesty Policy

- Academic dishonesty has no place in a university
- The Watson School has an academic honesty code.
- It is your responsibility to read and understand it.
- When in doubt, ask beforehand!

http://www.binghamton.edu/watson/about/honesty-policy.pdf

Academic Honesty on Assignments

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

Academic Dishonesty on Assignments

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) as another student

Academic Honesty on Tests and Exams

- Tests and exams will be open book to remove the temptation to look at the book.
- Test and exam questions will be relatively unique and specific to avoid quick lookup
- Answers will be compared with other students and material available on-line
- I depend on YOUR integrity to provide your own answers!

- (and it's hard to cheat on the oral exam)

Exams and Assignment Deadlines

- No make-up exams
  - Very few exceptions – medical emergencies, technical problems, etc.
  - The earlier we know about problems, the more flexible we can be
- Assignment Deadlines are firm
  - Some flexibility if you are 10 or 15 minutes late
  - Late penalties will be assessed

- Please plan other commitments around these deadlines!
  - Travel, interviews, etc.

Evaluation and Grading

Tests - 15% each	30%
Lab Average	15%
Assignment Average	15%
Final Project	10%
Graded Quizzes and Participation	10%
Final Exam	20%

There is no predefined average number to letter mapping!
Letter Grades depend on comparisons with students in previous and current semester, difficulty of tests, etc.

Dealing with Covid-19

- Stay healthy! Mask up. Social Distance.
- Respect privacy
  - Depend on health officials to keep us safe!
- Support each other!
  - Be thoughtful, accommodating, respectful, and honest
- Let me know if I can help