CS 5004: OBJECT ORIENTED DESIGN AND ANALYSIS

SPRING 2022

COURSE OVERVIEW

Instructor: Divya Chaudhary
d.chaudhary@northeastern.edu

Material Credits: Tamara Bonaci

Northeastern University
Khoury College of Computer Sciences

CS 5004 – OBJECT-ORIENTED DESIGN AND ANALYSIS

- Instructor:
  - Divya Chaudhary
  - Office hours:
    - TBD PT using Northeastern Zoom
    - By appointment

CS 5004 – OBJECT-ORIENTED DESIGN AND ANALYSIS

- TAs:
- Joseph Gramstad
- Shruti Aggarwal
- Bizhu He
- Tiffany Lastimosa
- Sheng Chen

- Course material: Canvas
- Course discussion board: Piazza
- Course assignment submission: Khoury GitHub
- Course grades: Canvas

CS 5004, SPRING 2022

- **CS 5004, Spring 2022 - teaching modality:**
  - **In-person lectures every Tuesday from 5:15-8:35 pm PT**
    - 401 Terry Ave N, classroom 106
    - I will do my best to make them available through Canvas + Panopto
  - **Hybrid labs:** on Wednesday from 6:30-8:00am PT in person and on Northeastern Zoom
  - **Lecture material** (lecture notes, reading notes, additional reading and videos) posted Canvas, under the specific weekly module
    - Relevant code using the course GitHub organization
    - Lab and homework assignments, and relevant additional material will be made available through Canvas + GitHub under the specific weekly module

WHAT IS CS 5004, SPRING 2022?

- Intensive tour of class-based programming design and principles
- Intended for students in the ALIGN MS in CS program
- Reviews typical object-oriented concepts
- Provides a deeper understanding of object-oriented design using UMLs
- Examines the relationship between algorithms and data structures
- Emphasizes testing, specifically unit testing of components
- Course goals:
  - Familiarity with the concepts and practice of object-oriented design
  - Design of abstractions that support the software management and reuse

CS 5004, SPRING 2022 – COURSE OUTCOMES

- At the end of this course, you should be able to:
  - Design an object-oriented solution to small and moderately sized problems
  - Implement a given object-oriented design in the Java programming language
  - Generate appropriate documentation for developed solutions
  - Design unit tests for a given component and implement them using the JUnit testing framework for Java programs
  - Create, refine and express a design in graphical notation such as UML diagrams
  - Explore existing documentation to describe and use existing libraries and frameworks

(EXPECTED) COURSE PROGRESSION

- **Week 1:**
  - **No lectures or labs.**
- **Week 2:**
  - **Lecture 1:** Getting Started. Data definitions and testing in Java.
  - **Lab 1:** Setting up the development environment and getting started.
- **Week 3:**
  - **Lecture 2:** Inheritance. Javadoc and UML diagrams. Code style. Equality. Exceptions.
  - **Lab 2:** Inheritance. Code style, Javadoc, and UML diagrams. Testing and exceptions.
- **Week 4:**
  - **Lecture 3:** Inheritance vs. composition. Interfaces.
  - **Lab 3:** Inheritance vs. composition. Interfaces.
- **Week 5:**
  - **Lecture 4:** Abstract Data Types (ADT). Java collections.
  - **Lab 4:** ADTs. Java collections.

(EXPECTED) COURSE PROGRESSION

- **Week 6:**
  - **Lecture 5:** Polymorphism. Overloading and overriding. Casting.
  - **Lab 5:** Polymorphism. Overloading and overriding. Casting.
- **Week 7:**
  - **Lecture 6:** Recursive data structures. Summary of OOD principles.
  - **Lab 6:** Recursive data structures. Midterm prep.

Week 8:

- **Midterm.**
- **No lab.**

- **Week 9:**
  - **Spring break.**
- **Week 10:**
  - **Lecture 7:** Generics in Java.
  - **Lab 7:** Generics in Java.
- **Week 11:**
  - **Lecture 8:** Java I/O. Java CLI. Regular expressions
  - **Codewalk 1.**

(EXPECTED) COURSE PROGRESSION

- **Week 12:**
  - **Lecture 9:** Java Collections framework. Interfaces Iterable and Comparable. Inner and nested classes
  - **Lab 8:** Java Collections Framework**.** Interfaces Iterable and Comparable.
- **Week 13:**
  - **Lecture 10:** Design of programs. MVC architecture.
  - **Codewalk 2.**
- **Week 14:**
  - **Lecture 11:** Design patterns and design principles. Functional programming in Java.
  - Optional final exam review.
- **Week 15:**
  - **Lecture 12:** Functional programming in Java. Networking in Java.
- **Exam week:**
  - **Final exam.**

COURSE LOGISTICS:

- Your work in this course will be graded based upon:
  - Homework assignments: 70%
  - Midterm: 10%
  - Final exam: 10%
  - Codewalks: 5%
  - Lab participation: 5%

COURSE LOGISTIC - HOMEWORK:

- Ten programming assignments
  - Six individual assignments
  - Four team / group assignments
- All assignments should be submitted using Khoury GitHub

Hard, long and time-consuming –
please start early!

COURSE LOGISTIC - EXAM:

- Two exams:
  - **Exam 1 – in week 8** – details TBD
  - **Exam 2 – final exam week** – details TBD

COURSE LOGISTIC – LABS AND LAB PARTICIPATION:

- Eight labs in this course
- Room: 225 Terry Ave, Classroom 402 – 6:30- 8:00 pm (Wednesday)
- You are expected to submit your lab work, and you will be graded based upon completeness (pass/fail grade)
- Goals:
  - Illustrate concepts covered in the lectures in a more hands-on style
  - Help with the next programming assignment

  Early labs - details related to:
  - Running and testing Java programs
  - Code management and version control
- Later labs - focus on design questions, and good Java programming practice

COURSE LOGISTIC – CODEWALKS:

- Two codewalks in this course (in the second part of the semester)
- Codewalk - a 15-20 minutes long presentation of your design approach, your code and your tests

COURSE MATERIAL

- No official textbook
- Good additional material:
  - **Java Tutorial** (https://docs.oracle.com/javase/tutorial/)
  - **How to Design Programs (2nd Edition)** by Felleisen, Findler, Flatt, and Krishnamurthi. Draft, 2010. (link: https://htdp.org)
  - **How to Design Classes** by Felleisen, Findler, Flatt, Gray, Krishnamurthi, and Proulx. Draft, 2012. (link: https://felleisen.org/matthias/HtDC/htdc.pdf)
  - **Absolute Java (6th Edition),** by Walter Savitch, 2016. (link: https://www.amazon.com/Absolute-Java-6th-Walter-Savitch/dp/0134041674)
  - **Effective Java (3rd Edition),** by Joshua Bloch, 2008.
  - (link: https://www.amazon.com/Effective-Java-Joshua-Bloch/dp/0134685997)
  - **Design Patterns: Elements of Reusable Object-Oriented Software**, by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, 1995.
  - (link: https://www.amazon.com/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612)
  - **UML Distilled:** A Brief Guide to the Standard Object Modeling Language, by Martin Fowler, 2003. (link: https://www.amazon.com/UML-Distilled-Standard-Modeling-Language/dp/0321193687)

COURSE MATERIAL

- Please "attend" lectures and labs because:
  - Lectures and labs will likely cover more than provided in the notes, and the provided references
  - Lectures will focus on "big-picture" principles and ideas
  - Labs will show you practical stuff and tricks you will need for homework assignments
  - Your colleagues will likely start interesting discussions during lectures and labs

LATE TURN-IN POLICY

- All assignments are due by 11:59pm PT on the assigned date
- Late assignments will (generally) be dropped 5% per calendar day, and no submissions will be accepted after 14 days
- *But, times are tough, and a little bit of flexibility could go a long way, so if you have a reason for delay (e.g., scheduling issues, childcare/ elderly care, job, health- related issues), please come and talk to me*

COLLABORATION AND ACADEMIC INTEGRITY

- General rule: you are encouraged to discuss all problems with us and with your classmates, but we expect everything that you submit to be your own work (code)
- That means that you should not share code, or reuse someone else's code (including the code possibly used in earlier iterations of this course, or the code that you may have found online)
- Any code that is identical, or similar to the existing code that we are already aware of will be penalized, and reported to the appropriate University authorities

TIPS FOR SUCCESS

- You cannot learn everything you need to know in lectures and/or through assignments
- You will also want to:
  - Read the assigned material
  - Attempt to solve additional problems
  - Attend lectures
  - Talk to the course staff
  - Keep up

YOUR QUESTIONS

?

Class Introductions

What is your name?

What was your undergrad degree?

What is your goal after you graduate?

Bonus Tell us a fun fact about yourself.