

CS 3354: Object-Oriented Design and Programming Spring 2017

Section 251
Section 252

Instructor: Dr. Jill Seaman
Comal 307G
js236@txstate.edu

Class Website: <http://www.cs.txstate.edu/~js236/cs3354>

Office Hours: M, W: 3:30pm – 4:30pm
T, R: 1:30pm – 3:00pm
and by appt.

Meeting Time/Place: Section 251: MW 2:00PM-3:20PM DERR 234
Section 252: MW 11:00AM-12:20PM DERR 240

Open Labs: DERR 231: Linux Lab
MCS 590: Windows Lab

Textbook: Object-Oriented Design and Patterns, by Cay S. Horstmann, John Wiley & Sons, 2nd edition, 2005. ISBN: 9780471744870 **REQUIRED.**

List of required readings and extra reference material: See the class website.

Prerequisites: Grade of C or better in CS 2308

Course Description: The course covers object-oriented design principles and programming for students with prior programming experience. The topics include inheritance and polymorphism, object-oriented design process, UML diagrams, design patterns, exception handling and multithreading. Students will design and implement programs in Java.

Grading:

Class participation:	5%
Exercises and Programming Assignments:	30% (6 total)
Midterm:	30% Wed, March 8
Final Exam (comprehensive):	35%
section 251:	Mon, May 8, 2:00pm - 4:30pm
section 252:	Mon, May 8, 8:00am - 10:30am

Class Participation: There will be some in-class activities. They may or may not be announced ahead of time.

Exercises and Programming Assignments: This portion of your grade is based on written homework assignments and programming assignments. The homework assignments involve drawing models and providing some written explanations. The programming assignments involve developing programs in Java. **You may do this work in pairs (with a partner) or by yourself (solo).**

Makeup Policy: Exercises and programming assignments cannot be made up. Exams may be made up in exceptional circumstances, with approval from the instructor.

TRACS: We will use the TRACS website for the following:

- Grades (Gradebook2 tool)
 - Programming assignment submissions and feedback (Assignments tool)
 - Resources (code you can use in your programming assignments)
- Everything else will be on the class website (including lecture presentations).

Withdrawals/drops: You must follow the withdrawal and drop policy set up by the University. You are responsible for making sure that the drop process is complete.

http://www.registrar.txstate.edu/registration/dropping_or_withdrawing

Last day to drop with automatic W: March 28, 2017.

Notifications from the instructor: Notifications related to this class will be sent to your Texas State e-mail account. Be sure to check it regularly.

Classroom Behavior: The main rule is to not disrupt or distract other students during class.

Academic Honesty: You are expected to adhere to the University's Academic Honor Code as described here: <http://www.txstate.edu/honorcodecouncil/Academic-Integrity.html>.

- **All assignments may be done in pairs or alone.** Each pair/individual must **write their own code.**
- Do not include code (or other materials) obtained from the internet in your assignments (except what is provided or allowed by the instructor).
- **Do not email your program to anyone (except your partner or the instructor)!**

The penalty for submitting a program that has been derived from the internet or any other non-approved source will be a 0 for that assignment.

Accommodations for students with disability:

Any student with needs requiring special accommodations should contact the office of disability services at the LBJ student center. Students who qualify for extra time for exams must take their test with ATSD and must schedule their test at the same time the test is given in class.

Course Objectives:

At the end of the semester the student should be able to:

1. Design, implement, test, and debug programs in an object-oriented programming language: Java.
2. Describe the unique features of Java.
3. Read and write Java programs that use generic types and data types from the Java Collections library.
4. Describe the concepts of inheritance and polymorphism and incorporate them into Java programs using abstract classes and interfaces.
5. Describe how the class mechanism supports encapsulation, information hiding, and interfaces.
6. Develop programs using multiple classes and composition.
7. Describe the semantics of exception handling in Java, and use it to write reliable Java code.
8. Read and write Java programs that use persistence (serializable objects).
9. Read and write Java programs that use threads to implement concurrency.
10. Describe and apply the Object-oriented design process to design a Java program.
11. Read, design, and draw the following models using the Unified Modeling Language (UML):
 - Class diagrams
 - Sequence diagrams
 - State diagrams
12. Write Java code that implements the designs specified by UML diagrams.
13. Describe the following Design Patterns and create UML designs using them, and implement the designs in Java programs.
 - Adapter
 - Strategy
 - Command
 - Composite
 - Observer
14. Determine the proper design pattern for a given problem.
15. Use Javadoc to specify the interface (API) of Java objects.
16. Understand and apply event-driven programming principles by developing programs with a graphical user interface, using objects from the Java Swing library.