CS 4354: Object-Oriented Design and Implementation
Summer II 2016

Section 751

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

Class Website: http://www.cs.txstate.edu/~js236/cs4354

Office Hours: MTWR: 11:45AM – 12:45PM
and by appointment.

Meeting Time/Place: MTWRF 10:00AM-11:40AM Comal 201

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 Readings on the class website.

Prerequisites: Grade of C or better in CS 3398

Course Description: An in-depth study of object-oriented design and implementation issues with emphasis on understanding the life cycle of object-oriented software, Unified Modeling Language, inheritance and polymorphism, designing remote and persistent objects, and exception handling. In-depth study of Java object-oriented language. Java will be used for implementing the exercises.

Grading:

Attendance:	expected	
Exercises and Programming Assignments:	40%	
Midterm:	25%	Tues, July 26
Final Exam (comprehensive):	35%	Thurs, Aug 11, 11:00AM

Attendance: I record attendance every day and I expect you to be in class every day. If you are unable to attend, you are responsible for the material and announcements that were covered that day.

Exercise 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 will do this work in pairs (with a partner).**

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 programing 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: July 29, 2016.

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. Please do not arrive late or leave early (without notifying the instructor).

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

- **All assignments are to be done in pairs.** Each pair 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 a special needs requiring special accommodations should inform me during the first week of class. The student should also contact the office of disability services at the LBJ student center.

Course Objectives:

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

1. Read and write code in a non-C++ language (Java)
2. Design, implement, test, and debug programs written in Java.
3. Describe the concepts of inheritance and polymorphism and incorporate them into Java programs.
4. Describe the semantics of exception handling in Java, and use it to write reliable Java code.
5. Read and write Java programs that use threads to implement concurrency.

6. Read, design, and draw the following models using the Unified Modeling Language (UML):
   - Use case diagrams
   - Class diagrams
   - Sequence diagrams
   - State diagrams

7. Use GRASP (General Responsibility Assignment Software Patterns) to determine what attributes and operations should go in each class in a UML class diagram.
8. Write Java code that implements the designs specified by UML diagrams.

9. Describe the following Design Patterns and create UML designs using them, and implement the designs in Java programs.
   - Adapter
   - Strategy
   - Decorator
   - Command
   - Composite
   - Observer
   - Iterator
   - Facade
10. Determine the proper design pattern for a given problem.

11. Use Refactoring to improve the maintainability of Java programs
12. Use JUnit to perform unit testing on Java code.
13. Use Javadoc to specify the interface (API) of Java objects.