CSC 3380
Aymond

Assignment

- Read Syllabus
- Read the LSU Code of Student Conduct Handbook section 10.1

Section 1

1/15/2020

Introduction

- Dr. Patti Iles Aymond
- paymond@lsu.edu **(By far, the best way to reach me)**
- (225) 578-4359
- 3270A Patrick F Taylor Hall
  - Monday & Wednesday
    - 1PM – 3PM
  - By Appointment

Course Description

- Advanced object oriented software development with an emphasis on the use of the unified modeling language as a design tool
- Course Objectives
  - Describe, compare and contrast common principles for **OO analysis and quality design**
  - Describe the **characteristics of quality software systems**
  - Describe the relative advantages and disadvantages among several **major process models**
  - Apply **fundamental principles of Object Oriented (OO)** programming
  - Formulate models of a design plan by utilizing **the Unified Modeling Language (UML)**
  - Classify **common patterns in OO design**
  - **Create a solution** using an appropriately selected design pattern
  - Apply communication skills necessary to **work in a collaborative environment on a team-based project**
  - Apply skills in **technical writing**
  - Apply communication skills necessary to participate in an **oral presentation and project demonstration**

Course Layout

- Lectures
    - Course lectures will be 1 hour and 20 minutes long, twice per week
    - All assigned reading is expected to be completed before class
    - A portion of class time will be devoted to project work, as time allows
    - Game plan:
        - Fundamental design principles, including system and software architecture
        - Design Patterns
- Homework Assignments
    - Homework assignments will be made available in Moodle.
    - These are to be completed and uploaded to Moodle, as instructed.
    - All UML assignments must be implemented using Enterprise Architect, unless otherwise specified.
- Class Project
    - Collaborative development effort
    - Real-world problem
- Exams
    - Midterm/Final
    - Not cumulative

Course Layout: Homework Assignments

- Approximately 6 homework assignments
- All UML homework assignments must be completed in Enterprise Architect, unless otherwise stated
- Enterprise Architect
  - Installed on LSU CSE lab computers
  - 1-week to complete homework assignments: don’t wait until last minute

Course Layout: Class Project

- Students will work together in a collaborative environment to tackle a complex problem within the problem domain posed by the instructor
- Students are expected to work together, divvy up responsibilities, and share resources in the implementation of the project
- Every student is expected to do their fair share of work on the project. A student's project grade will be proportionally adjusted based on level of effort expended on the project
- **A student cannot successfully complete this course without substantive contribution to the team project**

Course Layout: Exams

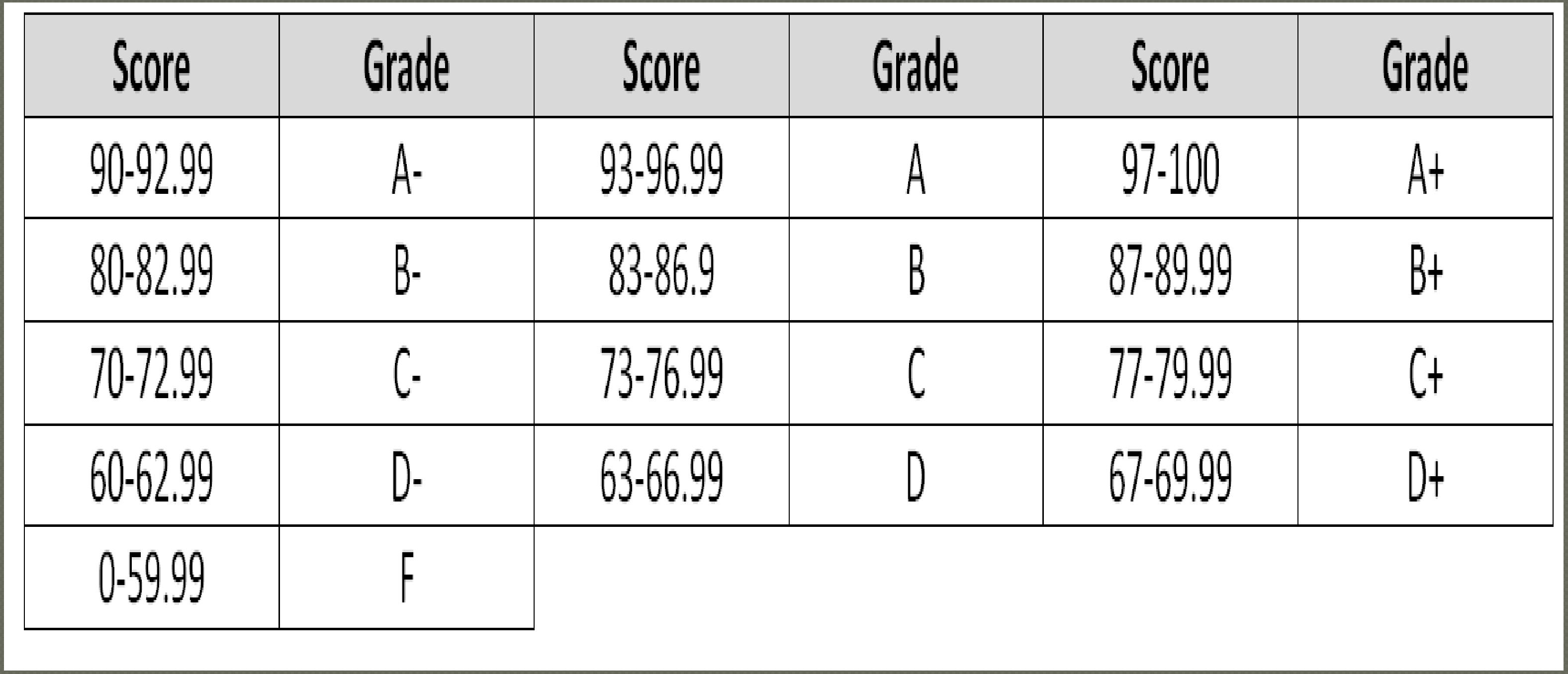
- There will be one, 80 minute midterm exam and a two hour final exam
- The final exam is not cumulative
- Arrangements for a make-up test must be made prior to the test. The instructor will be following LSU policy PS-22 with regards to valid reasons for missing an exam

STUDENTS MUST PRESENT THEIR LSU ID WHEN TURNING IN A COURSE EXAM

Evaluation

- Grading will be based on homework assignments, the class project, and two exams
  - Midterm Exam: 20%
  - Final Exam: 20%
  - Homework & Class Attendance: 20%
  - Class Project: 40%

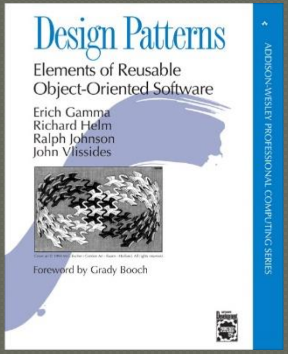
Grading Scale



Score	Grade	Score	Grade	Score	Grade
90-92.99	A-	93-96.99	A	97-100	A+
80-82.99	B-	83-86.9	B	87-89.99	B+
70-72.99	C-	73-76.99	C	77-79.99	C+
60-62.99	D-	63-66.99	D	67-69.99	D+
0-59.99	F				

Course Textbooks

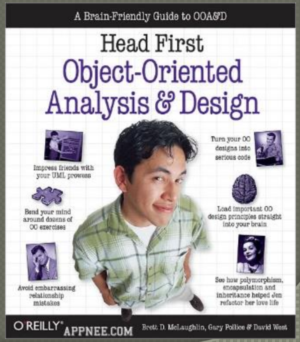
- Required
  - *Design Patterns: Elements of Reusable Object-Oriented Software* by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides


Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

- Recommended
  - *Head First Design Patterns: A Brain-Friendly Guide* by Eric Freeman, Elisabeth Robson

A Brain-Friendly Guide
Head First
Design Patterns
O'REILLY
Eric Freeman & Elisabeth Freeman
with Kathy Sierra & Bert Bates

- Additional Reading
  - *Head First Object-Oriented Analysis and Design* by Brett D. McLaughlin, Gary Pollice, Dave West


A Brain-Friendly Guide to OOA&D
Head First
Object-Oriented
Analysis & Design
O'REILLY
APPNEE.COM
Brett D. McLaughlin, Gary Pollice & David West

Class Policies: Class Attendance

- Students are required to attend class
- Class time will sometimes be used as project time; if students are not in class, their team will be adversely affected by their absence
- Class attendance may be taken and will count as a homework assignment grade
- Attendance WILL be taken on project kickoff and project presentation days

Class Policies: Moodle

- The course Moodle will be the official avenue for communication between the instructor and teaching assistants and students
- Moodle will be used for
  - providing general course information
  - making announcements
  - making assignments
  - turning in assignments
  - posting grades
- Report Moodle access issues to the Instructor ASAP

Check Moodle often for new or updated information.

Class Policies: Communication

- All electronic communication between students and the instructor and teaching assistants should maintain a **professional** decorum:
  - Address the recipient properly
  - Use whole words
  - Use complete sentences
  - Use proper spelling and grammar
- Be sure to include enough **detail** of the problem so that you form a well-thought out question. The more detail you can give as to your problem, the more likely the instructor or teaching assistant will be able to help you

Class Policies: Distractions

- Please silence your phones and put them away at the beginning of every class.
  - If you receive an urgent call or text during class, please take the device into the hall to complete your call/text
- If you must enter class late or leave early, please do so as quietly as possible
- Please avoid activities that will distract your classmates:
  - Talking
  - rustling papers
  - moving around
- The instructor reserves the right to expel a student from the lecture room for distracting, disruptive, or disrespectful behavior

Class Policies: Grading Change Requests

- Concerns about grades must be addressed **within one week** after the graded work is made available. Thereafter, all grade book entries are final
- Grade change requests must be submitted to the instructor, in writing, and must include the following:
  - Name of student
  - Date of the assignment/project/exam
  - Date of submission
  - Course number
  - List of the items that need to be corrected along with a concise reason as to why the grade change is needed.
  - Original assignment/test/quiz (not a copy)
- Grade change request decisions are at the discretion of the instructor and will be returned as the instructor has time to complete the regrade
- Final exam and final grade change requests can be submitted up to two weeks after the final exam

Class Policies: Due Dates

- **All work intended for grading must be submitted on time**
  - Any work not submitted before the cut-off period is not graded
- Homework assignments assignments will be submitted via assignments on the course Moodle.
  - Moodle is unforgiving, so do not wait until the last minute to upload your assignment to Moodle.

Class Policies: Missed Assignment/Exam

- A grade of 0 is awarded for missed assignment/exam in the absence of a valid excuse, as determined by the instructor
- In the unusual circumstance that you must miss an assignment or exam due to medical reasons or other unforeseen emergency, you need to
    - notify the instructor as soon as possible
    - provide sufficient documentation to verify the claim
- The instructor will be following LSU policy PS-22 with regards to valid reasons for missing an exam.
- If the instructor deems that the excuse is valid and sufficiently documented, the instructor will determine how the missed work will be made up, depending on the circumstance

LSU policy PS-22

- Valid reasons for absences include:
  - Illness
  - Serious family emergency
  - Special curricular requirements such as judging trips or field trips
  - Court-imposed legal obligations such as subpoenas or jury duty
  - Military obligations
  - Serious weather conditions
  - Religious observances.

https://www.lsu.edu/policies/ps/ps_22.pdf

Class Policies: Special Accommodations

- Students who have a disability that require accommodation(s) should make an appointment with the Office of Disability Services (Phone (225) 578-5919 or TDD: (225) 579-2600) to discuss their specific needs and present a letter from the ODS informing the instructor of their needs.
- All such matters, by University regulations, are strictly confidential.

Class Policies: Collaborative Work

- All class exams and homework assignments must be the independent work of the student
  - **CODE SHOULD NEVER BE COPIED FROM ANY SOURCE**
- Students are encouraged to work together on homework assignments, but the work you turn in must be your own. In other words, you can discuss problems and approaches to your problems, but **you should never share answers and source code (including eap files)**.
- Students are required to abide by the LSU Code of Student Conduct Handbook (see the syllabus for important details).

Important Dates

- Midterm Exam: Wednesday, March 4, during schedule class time
- Final Exam: Saturday, May 9; 7:30AM – 9:30AM

- Project Kickoff: Monday, January 27
- Project Architecture Presentations: Monday, March 2
- Final Project Presentations during class: Wednesday 4/22, Monday 4/27, and Wednesday 4/29

- SUBJECT TO CHANGE DUE TO NATIONAL CHAMPIOINSHIP FOOTBALL GAME:
  - Wednesday, January 22 – Final day to drop without a W (4:30 PM deadline)
  - Monday, April 6 - Final day for dropping classes (4:30 PM deadline)
  - Monday, April 6 - Final day to request rescheduling a final examination when three examinations are scheduled in 24 hours

- Check Moodle often for class schedule updates

Communication Across the Curriculum (CxC)

- This is a certified Communication-Intensive (C-I) course which meets all of the requirements set forth by LSU's Communication across the Curriculum program, including
  - instruction and assignments emphasizing informal and formal [mode 1] and [mode 2];
  - teaching of discipline-specific communication techniques;
  - use of feedback loops for learning;
  - 40% of the course grade rooted in communication-based work; and
  - practice of ethical and professional work standards.
- Students interested in pursuing the LSU Communicator Certificate and/or the LSU Distinguished Communicator Medal may use this C-I course for credit
- For more information about this student recognition program, visit www.cxc.lsu.edu

Class Project

- On project kickoff day, the instructor will provide a general problem domain within which the project must apply
  - The team will craft a specific solution within that domain
- The class will be partitioned into teams of 6
  - Students may propose teams, but the instructor reserves the right to shuffle teams to ensure teams of size 6
- While teams are expected to equitably divvy responsibilities among team members, all team members are required to implement system and/or software designs in Enterprise Architect
- All team members are required to participate in the in-class final project presentation

Class Project: Team Leadership

- The team must select a leader that will take on the responsibility of coordinating the team efforts
  - The team leader is expected to shoulder leadership responsibilities in addition to project responsibilities
  - If the team leader provides good project leadership, they will get extra project points for their leadership efforts
  - If the team leader fails to effectively lead the team, the team leader can lose project points
  - The team may choose to change leadership for different Milestones, upon instructor approval

Class Project: Grade Calculation

- Project grades are calculated as follows:

$$P \times \sum_i (Weight_i \times Grade_i \times PercentContribution_i \times NumTeamMembers \ \times L_i)$$

- Where *i* ∈ {*Milestone 1, Milestone 2, Milestone 3, Milestone 4, Final Presentation, Post Mortem*}
  - $Weight_{Milestone\ 1} = 0.1$
  - $Weight_{Milestone\ 2} = 0.25$
  - $Weight_{Milestone\ 3} = 0.25$
  - $Weight_{Milestone\ 4} = 0.25$
  - $Weight_{Final\ Presentation} = 0.1$
  - $Weight_{Post\ Mortem} = 0.05$
- $Grade_i$ is the grade awarded to the team project for Milestone *i*
- $PercentContribution_i$ is the percentage that an individual team member contributed to the milestone
  - It is expected that every team member will contributed approximately the same amount to each milestone, but this factor corrects for the situation when contribution is not equitable within a team
  - This factor is determined by a survey of team members
- $Leadership\ Factor_i$ is the leadership offset for Milestone *i* (either 1.15, 1.0, or 0.85)
- P is the presentation factor
  - 1, if the student provided an in-class presentation
  - 0.9, if the student did not provide an in-class presentation

Class Project: Milestone Dates

- Project Kickoff
  - In class, Monday 1/27
- Milestone 1: Stories & Requirements
  - Due: 11PM Tuesday, February 4
- Milestone 2: Architecture Design
  - Due: 11PM Friday, February 21
  - In-class presentation: Monday, March 2 <tentative>
- Milestone 3: Component Design
  - Due: 11PM Tuesday, March 17
- Milestone 4: Working Prototype
  - Due: 11PM Tuesday, April 21
- Final Presentations
  - In class, 4/22, 4/27, & 4/29
- Post Mortem
  - Due: 11PM Friday, May 1

Challenges of Software Development

- Complexity
  - Software systems today are typically very large and very complex
- Longevity and Evolution
  - Systems are often in service for long periods of time
  - Being used for applications for which it was never intended
- High user expectations
  - Diversity of needs
  - Expectation for quality and security
  - Voodoo magic

Jia, Xiaoping. Object Oriented Software Development Using Java: Principles, Patterns, and Frameworks. Addison Wesley, 2003.

Desirable Qualities of Software Systems

- Usefulness
  - Adequately address needs
- Timeliness
  - Quickly developed and deployed
  - Continuous integration/continuous deployment
- Reliability
  - Perform as expected
- Maintainability
  - Can easily make corrections, adaptations and extensions
  - Flexibility
    - Easily changable
  - Simplicity
    - Anticipate and deal effectively with human error
  - Readability
    - Clarity and simplicity of design
- Reusability
  - Components can be repurposed for other applications
- User Friendliness
  - Intuitive use and access
- Efficiency
  - Efficient use of processing time, memory, and disk space

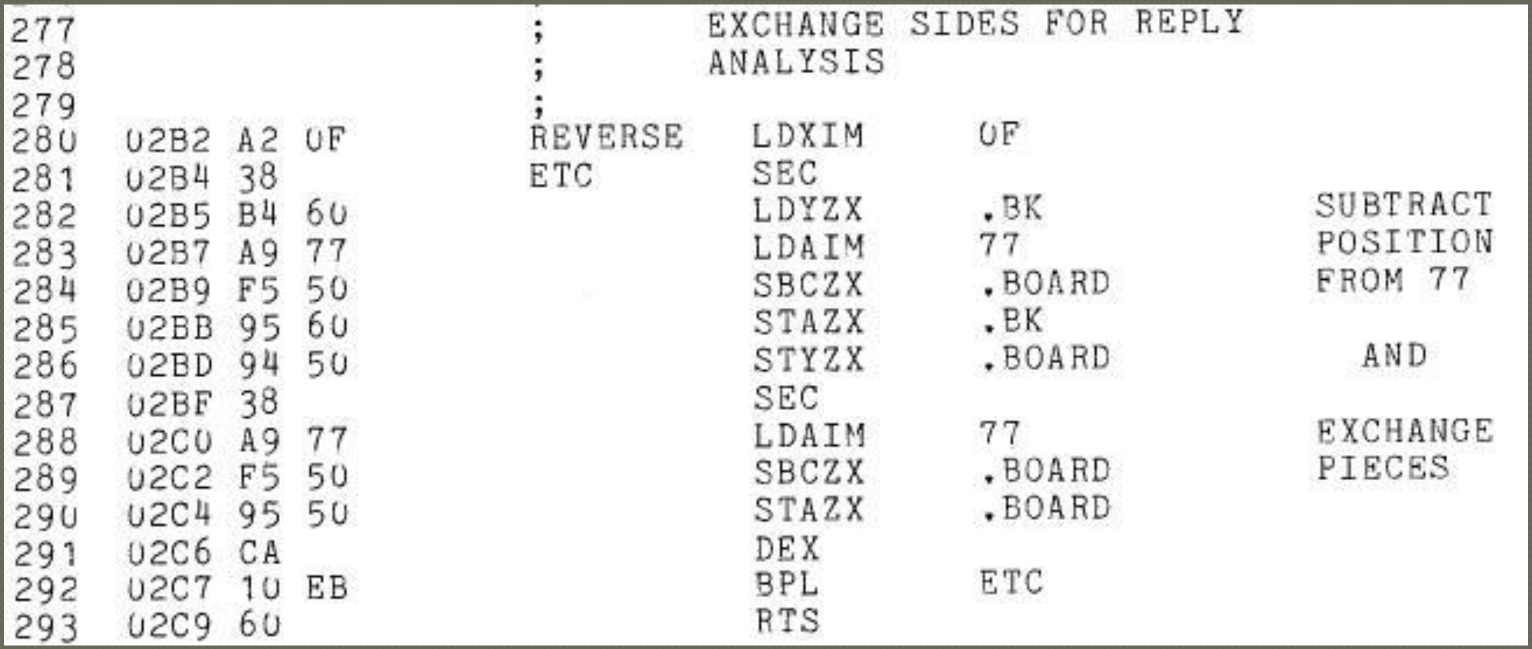
Jia, Xiaoping. Object Oriented Software Development Using Java: Principles, Patterns, and Frameworks. Addison Wesley, 2003.

Programming Paradigms

- A paradigm is a way of looking at the world
- A programming paradigm is a way of looking at programming
- To understand OO, we need to understand what came before

Pure Imperative

- "Every program is a list of instructions"
  - Data up front
  - Procedural logic
- Fosters spaghetti code
- Imperative Example: 6502 Assembly



```
                     ;         EXCHANGE SIDES FOR REPLY
                     ;         ANALYSIS
                     ;
02B2 A2 0F           REVERSE   LDXIM     0F
02B4 38              ETC       SEC
02B5 B4 60                     LDYZX     .BK          SUBTRACT
02B7 A9 77                     LDAIM     77           POSITION
02B9 F5 50                     SBCZX     .BOARD       FROM 77
02BB 95 60                     STAZX     .BK
02BD 94 50                     STYZX     .BOARD         AND
02BF 38                        SEC
02C0 A9 77                     LDAIM     77           EXCHANGE
02C2 F5 50                     SBCZX     .BOARD       PIECES
02C4 95 50                     STAZX     .BOARD
02C6 CA                        DEX
02C7 10 EB                     BPL       ETC
02C9 60                        RTS
```

Structure Programming

- “Programs are hierarchies of structures.”
  - If statements
  - While loops
  - Functions
  - Lexical variables
- Structured example: C

```
long  fibo( unsigned char i ) {
    long j;
    long a = 0;
    long b = 1;

    if( i == 0 ) { return 0; }

    for( j = 1; j < i; j++ ) {
           long new = a + b;
           a = b; b = new;
    }

    return b;
}
```

The Object-Oriented (OO) Paradigm

- Abstraction
  - Hidden Data
    - Implementation of Abstract Data Type (ADT) is irrelevant
    - *** **Class members are not (NEVER) accessed directly** ***
      - In C++, this means no public class members
- Encapsulation
  - Data and methods on that data are bundled together
    - A class defines the data implementation, access to the data elements, and methods that act on the data
- Inheritance
  - A class can take on the properties of another class
    - Creates the is-a relationship between the base class and the superclass
- Polymorphism
  - Derived objects (those of a class inherited from another) can behave differently
    - Interface of inherited methods remain the same, but may function differently