CS 417/505
Design Patterns

Preliminaries

Dr. Chad Williams
Central Connecticut State University

Agenda

- Get to know each other
- Walk through syllabus
  - Course description & objectives
  - CS 417 vs CS 505 expectations
  - Grading
    - Class participation
    - Assignments
    - Exams
    - Course Project
  - Course calendar

Getting to know each other

- Name
- Undergrad or Graduate student
- Prior experience with development outside of classroom
- What do you hope to get out of the class
- What sort of job do you want after graduation
- Something unique about yourself

Basics

- **My office:** MS 303
- **Phone:** 860-832-2719
- **Office hours:** TBD; and by appointment.
- **e-mail:** cwilliams@ccsu.edu
- **Course websites:**

  Blackboard
  - Assignments and grades

  GitHub (https://github.com/CCSU-CS417-505-F19/DesignPatternsCourseInfo)
  - Everything else – slides, examples, sample code, reference material, coding assignments will also be submitted via GitHub (more details to come on that)
- **Class times:**

  4:30-5:45pm Monday and Wednesday in MS 204

Text book and references

- **Optional book:**
  Gamma, Helm, Johnson, and Vlissides.
  "**Design Patterns: Elements of Reusable Object-Oriented Software**" (1994)
  informally known as "The Bible by the gang of 4" when it comes to design patterns.

What is this class about?

- Prior classes focused on functionality i.e. “create a program that does ***X***”
- This class focuses on designing solutions and implementing quality programs
  - Team development
  - Knowledge of advanced OO design and patterns
  - Ability to recognize when using a design pattern would be appropriate and why
  - Use UML to come up with a solution
  - Develop automated testing
  - Proper implementation
- These skills are highly sought after professionally, and are what will make you a ***great*** developer

Material covered

- UML design
- Git
- Coding best practices
- Automated testing
- Using abstraction to improve design
- Developing OO application frameworks
- Making better use of existing tools to make your lives easier and avoid reinventing the wheel

Topics

- Some already covered, but more depth
  - Inheritance
  - Encapsulation
  - Modularity
- Others that may have been visited but not fully appreciated…
  - Abstraction
  - Polymorphism
  - Message passing
  - Exception handling

Topics cont.

- Coverage of the following design patterns, their use and implementation
  - Chain of responsibility
  - Singleton
  - Template method
  - Strategy
  - Iterator
  - Factory
  - Composite
  - Delegation
  - Decorator
  - State
  - Abstract factory
  - Prototype
  - Builder
  - Visitor
  - Flyweight
  - State

Topics cont.

- Incorporating frameworks
  - Collections
  - Generics
  - Composite
  - Comparators
  - MVC
- Other key topics
  - Source control
  - UML design
  - Implementation of all patterns
  - Creating automated testing
  - Proper exception handling

Course outcomes

At the end of this course you should:

- Have an in depth understanding of coding in a team environment
- Be able to apply advanced object oriented design patterns to real world problems
- Understand how to use UML in complex RUP design: Object and classes, relationships, state and sequence diagrams
- Have an increased understanding of the key aspects of object-oriented design: abstraction, encapsulation, modularity, message passing, polymorphism, inheritance, exception handling

Course outcomes cont.

- Develop an advanced knowledge of the following design patterns: Singleton, Template, Strategy, Iterator, Factory, Delegation
- Know how to incorporate the following frameworks into an application design: Collections, Generics, Composite, Comparators, and MVC
- Understand unit and integration testing design and how to automate testing of each
- The course will culminate in a large final project that will require working on teams and choosing the right design patterns covered throughout the course
- **Result: Not only do you know how to develop and implement secure applications, you are also more marketable!**

CS 417 vs CS 505 expectations

- What is the difference between CS 417/CS 505?
  - Higher expectations on extent of mastery of material for graduate students.
  - Graduate midterm/exam more in depth than the undergraduate version
  - Final project requirements
    - Graduate teams will have to provide a component to be used by other teams and support its use/evolution in addition (more on this later)

Grading for the course

Letter grade will be calculated according to the following table:

A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F
95-100	90-94	87-89	84-86	80-83	77-79	74-76	70-73	67-69	64-66	60-63	0-59

Percentage of grade:	
Assignments	25%
Exams	30%
Final project	45%

- Basic philosophy:
  - In class work and assignments are your chance to learn, effort rewarded
  - Exams test that you learned from your mistakes
  - Final project demonstrates applied understanding

Participation and class work

- I take these very seriously
- **If you do not participate you will not receive a passing grade for this course**
- We will be doing a **lot** of in class group work
  - Simply being part of a group that does the work does not count as participating
  - It means fully engaging and contributing to your group's work

http://www.cs.ccsu.edu/~williams/classes/WhatDoesClassParticipationMean.pdf

Exams (30% of grade)

- Midterm (15%)
  - Wednesday 10/16
- Final (15% cumulative) will be at the university's scheduled time:
  - Monday December 9th 5:30-7:30pm

Development practice

- Real projects involve teams and evolving code based with multiple people making changes simultaneously – this class is designed to produce this same scenario
- **Require** teams for majority of assignments
- Proper use of source control (Git)
- Unlike many classes where you are evaluated just on the final deliverable, **expect to be evaluated on whether you followed good development practices along the way as well**

Git

- Source control is a critical part of any real world development environment and thus is a critical skill for you to know
- You will be taught Git, and it will be used extensively throughout this class
  - Collaboration
  - Workflow
  - Releases
- Both are proven and proper use of them on your projects and assignments is a requirement

Assignments (25%)

- Hands on problems based on lectures and reading

- Two types:
  Design
  In addition to having the correct design you must explain the rationale for your approach to get full credit. In the professional world being able to articulate your reasons is a critical component of selling your solution to others even if it is the right solution. The same can be said in this class.

  Programming
  Uncommented code will not receive full credit. Simply having code that works, but does not implement the specified design pattern is not acceptable. Proper use of source control is required for full credit.

- Start your work early!

Final project (45% of grade)

- **Must** be done in teams
- It will incorporate material covered throughout the semester – some elements specified as required, others you will choose
- Interim check points starting early in the semester contribute to this grade
- Last classes your group will present and demonstrate your version of the project for the class

Final projects

- Goals:
  - Learn how to effectively work with complexity of properly implemented and executed large scale multi-person team project (i.e. good development practices)
  - Demonstrate you know **proper** time to use patterns and how to implement them correctly
  - Make your project fun! The projects are designed to allow you to be creative as you want.
- **Warning this is a large project, leaving it for the last minute is a recipe for disaster**

Final project deliverables

- Interim milestones
  - Design document
  - Implementation checkpoints
- Final deliverable
  - Write up explaining what you implemented including screen shots
  - Write up of design patterns used **and why** they were chosen
- Well commented source code of your implementation
- Presentation slides
  - You will need to give an oral presentation and demonstration of your application to the class
  - Final projects will be due at the **beginning of class** Monday, December 3rd

Late work

- Late work assignments and project checkpoint deliverables are accepted with a penalty of **10% per day** late. Note 2am is same amount off as 11:59pm that day
- No work will be accepted after the final exam
- Be sure to keep pace and start work early to avoid getting into trouble
- For the final project presentation, if your group isn't ready to present Monday **at the beginning of class** it is 20% off and you can try again Wednesday

Attendance

- I expect students to attend class regularly not doing so **will** impact your grade
- For each absence over 3 will reduce your overall final letter grade by 1/4th (unless university excused)
- If you miss class it is **your responsibility** to get announcements and determine what you missed from your classmates
- In the event of a weather emergency that requires curtailment or cancellation of classes, listen to WTIC (1080 AM) or call (860) 832-3333

Common Sense

- No cell phones or texting doing so may result in me asking you to leave class in which case it will count as an absence for the day as well as impacting your participation grade
- Cheating will not be tolerated
  - Turning in someone else's work as your own
  - Allowing someone else to copy your work
  - Using a solution from a previous term or from the web
  - Plagiarizing
  - **Penalties for cheating go beyond just receiving a zero for the assignment and could result in penalties as harsh as failure of the course and an Academic Misconduct Report being filed**

Students with special needs

- Please contact me privately to discuss your specific needs if you believe you need course accommodations based on the impact of a disability, medical condition, or if you have emergency medical information to share. I will need a copy of the accommodation letter from Student Disability Services (SDS) in order to arrange your class accommodations. Contact SDS in Willard Hall, Suite 101-03 if you are not already registered with them. SDS maintains the confidential documentation of your disability and assists you in coordinating reasonable accommodations with your faculty.

How to succeed

- **USE OFFICE HOURS!**
  - Work through practice problems
  - Discuss some topic you are having difficulty with
  - Put simply, **if you are confused come see me**. While I have posted office hours, feel free to stop by anytime or make an appointment with me if you want to make sure I'm available at a specific time.
- **Do all of your assignments**. It is the best way to realize quickly if you are missing any important points so that you don't make similar mistakes on the mid-term and final exams.
- **Get involved** in lectures. Don't be afraid to ask for clarification or additional explanation, chances are if you are confused someone else is as well.
- **Pay attention to all of your feedback on homework and exams**

Keeping up with latest

- Blackboard will have
  - Announcements
  - Assignments
  - Link to GitHub page
  - Team sign up
  - Grades
- GitHub will have (*more on this next*)
  - Latest syllabus
  - Class slides/material
  - Useful links and tutorials
  - Class notes
  - Repositories for your assignments and final projects

GitHub for course material/submissions

- Course material
  - If slides or code examples are updated you are notified of changes and can ensure you have the latest
  - Also it is by far the best way to distribute code examples
- Course code version control
  - Can monitor to be sure you know how to use it properly
  - Ability to easily provide source code feedback
  - If there is an issue of an underperforming team member I can see exactly what their contribution is
- Practice, practice, practice!

“Homework” 0 due Wednesday 9/5

- Submit through Blackboard a **head shot** of yourself and **GitHub id**.
- Please make it close enough that I can clearly make out your face