Object-Oriented Programming
Course Syllabus

Course & Instructor Information

Course:	Lewis University, Object-Oriented Programming, CPSC-24500-LT1, Online, Spring 2018
Dates:	Tuesday, 16 January 2018 through 12 May 2018 per Academic Calendar [link]
Times:	TTh 2:00 to 3:15pm CST
Location:	AS 104A
Final:	Tuesday, 8 May 2018 from 1:30 to 3:30pm in our normal location (AS104A)
Instructor:	Eric Pogue
Email:	epogue@lewisu.edu …you can expect a response within 24 hours weekdays
Phone:	563-209-7280 (personal mobile)

Object-Oriented Programming Course Description

Students will learn to design and develop software using the object-oriented programming (OOP) techniques. Topics include encapsulation, inheritance, polymorphism, abstraction, and patterns. Students will learn how to use and libraries and software development kits (SDKs) to develop applications that provide data processing and visualization services. Students will also learn how to manage threads and networking connections in software they write.

In addition, students will learn the basics of developing software utilizing Agile processes, terminology, and development techniques.

Prerequisites:	CPSC 21000 Programming Fundamentals or consent of instructor
Credits:	3

Course Materials

Reference materials for this course will focus on class slides, notes, and online sources.

Textbook:	No textbook is required for this course.
Reference Materials:	Class slides, notes, and online sources will serve as your primary reference materials. These will be posted online and linked via Blackboard. New materials will be made available at one and two-week intervals.
Recordings:	Lecture sessions and important topics will be recorded and posted online… as long as this practice does not have a negative impact on class attendance or participation.

Student Learning Outcomes

On the successful completion of this course you will be able to:

- Understand the basics of various software delivery process and their implications for utilizing object-oriented (OOP) programming techniques
- Utilize Agile software delivery techniques and terminology to deliver assignments
- Effectively utilize key developer tools with an initial focus on Git and GitHub
- List and explain the key concepts of object-oriented development: encapsulation, inheritance, and polymorphism
- Write class definitions and create objects from them
- Declare and use special types of functions for classes, including constructors, accessors, and properties
- Define the following object-oriented patterns: Factory, Singleton, Delegation, and Model-View-Controller
- …
- Retrieving data from a website
- Write programs that use various collections
- …
- Perform input and output with JSON (or possibly xml) file streams and serialization
- Use an API as a reference when writing programs
- Build attractive, intuitive graphical user interfaces
- …
- Create a user interface in Java and C# (possibly C++)
- Describe how Java achieves cross-platform compatibility
- Distinguish among heavyweight and lightweight components
- Define callback function as it relates to event handling
- Respond to user events in Java and C# (possible C++)
- Describe how layout managers arrange components
- Understand how to write unit tests to verify the correctness of software modules

Class Assignments

Assignments for this course will take the form of quizzes, labs, programming projects, and participation.

Tests:	There will be no midterm or final tests for this course.
Quizzes:	Quizzes will be assigned at the end of one or two-week intervals.
Programming Projects:	Programming projects are the most important element of this class and will be assigned to be delivered in one or two-week intervals.
Participation:	Participation and engagement can be demonstrated in class, via Blackboard discussions, and through your class presentations/demos.

Programming projects are the most essential element of this course and will account for approximately 60% of the final grade. Leaving quizzes and participation to account for approximately 30% and 10% respectively.

Late Assignments & Partially Completed Assignments

Late assignments will not be excepted except under extreme circumstances. It is vastly preferable to turn in a partially complete assignment than to turn in a late one.

Similarly, it is vastly more beneficially to turn in an assignment that has 70% of the features working 100% than to turn in an assignment that has 100% of the features working 70%.

Grading Policies

Final course letter grade will be determined using the following approximate scale:

A	>= 90	C-	70-72.99
B+	87-89.99	D+	67-69.99
B	83-86.99	D	63-66.99
B-	80-82.99	D-	60-62.99
C+	77-79.99	F	< 60
C	73-76.99		

Drop and Withdrawal Deadlines

Information regarding important drop and withdrawal dates and policies can be found on the Bursar's webpage [link].

Plagiarism:

Copying work from each other or from the Internet will be punished harshly and appropriately. This includes viewing test, quiz, or assignments from current or previous class sessions. Measure of Software Similarity (MOSS) or similar software may be utilized to detect copied work. If it is determined that you have copied/plagiarized your work, you will fail the assignment and may have addition points deducted from your grade equal to the assignment maximum. If you do this twice, you will fail the course and potentially face additional disciplinary action. Integrity is expected and required.

Also see "Academic Honesty" below.

Closing Comments:

Nearly all modern, sophisticated software is created utilizing object-oriented programming techniques. I hope that this course will allow you to dramatically improve your programming skills and capabilities in the coming weeks. Please make sure you let me know when you're struggling so that I can help you be successful. Finally, be sure to take a moment and enjoy the journey.