CSC-133 :: Course Overview

Welcome to CSC-133 Sections 3 and 4 :: GROOP
(Object Oriented Programming with Graphics)

COURSE DATES | Aug 31 - Dec 11

This is (almost) a fully online course. Lecture content will primarily be delivered asynchronously. There will be synchronous discussion sections, as needed. These sessions will be announced on Canvas. Expect that these discussions will coincide with the normally scheduled meeting time.

We have sixteen (16) weeks and eight (8) modules to complete. Due dates are set for pacing. You will see a link to each semi-weekly module listed below. You can view the full list of modules via the **Modules**link on the left-side course navigation menu.

Course Overview

This course gives an introduction to object oriented program and computer graphics. Over the course of this semester you will work on a game project using an event driven and graphically oriented environment. You will be introduced to computer graphics topics and to advanced topics in object-oriented (OO) programming. The OO paradigm is used throughout, utilizing computer graphics as the vehicle for solidifying basic OO concepts, studying the implementation of event-driven systems, and for developing a thorough understanding of advanced OO concepts such as inheritance and polymorphism. Topics include fundamental concepts of object-oriented programming, software design patterns, graphic devices, line and surface drawing, simple 2D and 3D representation, and use of User Interface components. The presentation of this material is divided up in eight modules over three major course sections, or parts. Over the course of the semester you will build up a GUI based, event driven video game suitable for a mobile device.

Part 1: Object Oriented Programming

In the first part of this course we dive deeper into Object Oriented Programming. We discuss class association, composition, and aggregation. We talk how we use abstraction and encapsulation to organize code. We talk at length about the importance of hiding data and how to properly use accessors and mutators. We revisit inheritance and discuss implementation vs interface inheritance. We then apply these ideas in the context of abstract lasses and Java interfaces. We will study polymorphism, runtime typing, late binding and compare and contract Java and C++. Finally we will dive into design patterns.

Part 2: GUI Programming

In part two we will learn about GUI programming. We will study the concepts of Graphical User Interfaces, frameworks, and specifics of Java and some aspects of mobile development. We will talk about how to build up a GUI from containers and components and how we leverage layout managers. We then get into event driven programming. We will discuss associated design patterns, how we model events in Java, Java threads, the concepts of a listener. Finally we will learn about simple animation, collision detection, and other topics of interest related to implementing a simple 2D java game.

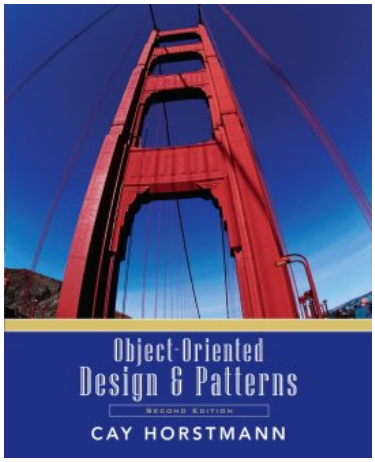
Part 3: Graphics Principles

In the third and final part of this course delve more deeply into principles of graphics programming. we will learn about dislays, colors, and transformations. We review some basic matrix and vector math and then use these tools to learn about transformations of graphics objects. We learn how to compose these transformations and how to write code to implement these transformations. Finally we will learn about local, world, and screen coordinates and how to display objects using windows, viewports, and the viewing transformation.

Course Work

The primary course assessment will come from four major labs and eight module quizzes. In addition, course participation, practice quizzes, and skill based homework will make up approximately twenty percent of your grade. As detailed in the schedule above, each module takes two weeks to complete. The labs will overlap the module completion time as they will put into practice skills learned in the modules.

Primary Textbook



The primary textbook for this course is **Object-Oriented Design & Patterns** by **Cay Horstman** . This textbook is only recommended. The choice to

purchase the book is yours. There will be recommended reading from this book, however, I will do my best to provide alternate readings in the support text.

Support Textbooks

In addition to the primary textbook, several other support texts are provided to give you a different perspective on some of the material. All of these additional texts are either open source or freely available on the web. For the texts below that are distributed with an open source license, they have been provided to you as a a download.

Textbook	Author(s)	License
Introduction to Computer Graphics An introduction to 2D computer graphics. This book has been edited from the original in that unneeded content has been removed. It includes a brief introduction to Gimp and Inkscape that students may find helpful when creating digital assets. However, those programs are not required for this course.	**David J. Eck**	**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**
Codename One Developer Guide This is the official guide for developing with Codename One. We will not be covering this document in depth. You will want to use it to help you get your environment setup and as a reference throughout the semester	Various, see document.	See document for distribution license terms and source material. This document is freely available on the web. Link is from the developers website.
Think Java 2nd Edition This **GreenTea Press** book on Java is provided as a reference for background information and as a fast read for students transferring in without a background in Java.	**Allen B. Downey** and **Chris Mayfield**	**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**
Java Java Java This open source book on Java is provided as a deeper reference for some of the more advanced OOP concepts that are not covered well by other books.	**Ralph Morelli** **Ralpeh Wald**	**Attribution 4.0 International (CC BY 4.0)**
Pre-Calculus This **OpenStax** textbook is provided as a reference in case you need a review of trigonometric functions or matrix operations. There will be reading assignments from this book, however, these assignments should be review and the content is considered to be pre-requisite material.	**Jay Abramson**	**Attribution 4.0 International (CC BY 4.0)**

Getting Started

Start by reading all of the material in the **Welcome module**, then make your way over to **Module 0 :: Prerequisites**to make sure that you are up to speed on the course prerequisite material. If your background so far has not covered all of this material, then be sure to check out the provided links to get up to speed as quickly as possible. While the prerequisite material has a slightly different presentation, the **Module Outline**page provides an overview of how the remainder of the content of the course is presented.

Course Schedule

Week 1/2 *Aug 31 - Sep 13* **OOP Concepts**	**Week 6/7** *Oct 5- Oct 18* **Graphical User Interfaces**	**Week 10/11** Event Driven Programs	**Week 15** **Module 3.1 Introduction to Graphics Principles**
Week 3/4/5 *Sept 14 - Oct 4 (3 Weeks)* **Association, Inheritance and Polymorphism**	**Week 8/9** *Oct 19 - Nov 1* **Design Patterns**	**Week 12/13** **Module 2.3 Interactive GUI Programming**	**Week 16** *Nov 30 - Dec 11* **Module 3.2 Graphics Operations and Programming**

Course Communication

Communication is the basis for quality interaction and participation in an online course. Demonstrating respect for one another is important as we build our learning community. Netiquette is the term used when referring to respectful and/or acceptable ways of communicating online. Please take a few moments to reference these **Netiquette Tips** as you get started in the course.

You can communicate with your facilitators and peers via the appropriate discussion forum, or via Canvas messaging (see Inbox icon in the global navigation menu to the left).

Course Announcements will be sent out regularly to keep you informed of course changes, new content, and any emerging issues arising from the current situation.

Students enrolled in my course ***must*** use Canvas messaging for all course related communication while the course is active. While a course is active, I do not respond to email from students enrolled in my course. After the Canvas course is closed you may send any university related communication to **my university email address** ***from your university email address***. I do not respond to university related communication from students sent from a non csus.edu address.