



Account



Dashboard



Courses



Calendar



Inbox



History



Online Teaching

Resources



Student

Technology

Resources



Library



Help

Summer 2023

Level 0.3 - Course Information

Menu

Here is the menu for you to navigate in this long document.

- [Title](#)
- [Information](#)
- [Requirements](#)
- [Objective](#)
- [Activities](#)
- [Schedule](#)
- [Policy](#)

[Course Evaluations](#)

+++ Warning +++

This is a hard course. It is programming intensive!

Many students failed this course as they start assignments late and cannot finish them on time.



Course Title

Official Title: CSC133 – Object-Oriented Computer Graphics Programming

My Course Title: CSC133 - Application Development with Object-Oriented Programming

Although the official course title is "Object-Oriented Computer Graphics Programming," the core of this course is not computer graphics. Computer graphics are usually not considered as a required topic for Computer Science students. This title is just for fulfilling the program requirements. In fact, I believe the title should be "Application Development with Object-Oriented Programming." We designed this programming-intensive course for students to gain experience in having a complete and large-scale application development from scratch. Students will work from reading specifications, designing code, building GUI, creating interactive systems, and finally, code release. It helps students enhance their programming skills, experience real programmer life, and employ learned computer science theory in application development, such that student can prepare for their senior projects and future career.

Course Information

Course Code: CSC 133

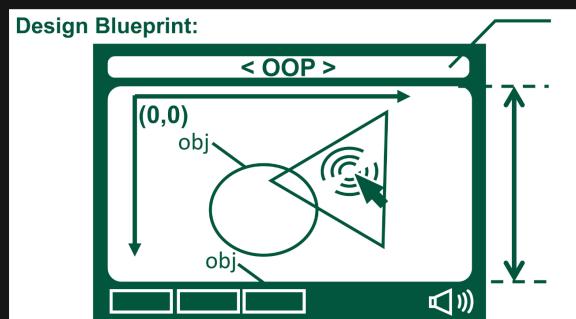
Title: Application Development using Object-Oriented Programming

Type: Required course of Computer Science Program

Description: A computer science student should have the ability to develop an application based on provided requirements. This course is designed to help students to gain experience in programming using object-oriented (OO) languages and application development tools. Coding design using OO concepts and software design patterns; Fundamental elements for application development such as graphical user interface, interactive, media, and general 2D graphics; Programming assignments to enhance OO programming skill and emphasizing complete applications development.

List of Topic: The following lists the topics that will be included in this course:

1. Object-Oriented Programming
2. Design Patterns
3. GUIs and Event-driven Programming
4. Interactive Applications
5. 2D Coordinate and Transformation
6. Application System



The overview image of CSC133. You can see design, OOP, GUI, sounds, graphical objects, interactive, and coordinate systems, which are the major topics in this course.

Requirements:

Course Prerequisite:

- CSC 130 – Algorithms and Data Structures (grade of C- or better)
- CSC 131 – Introduction to Software Engineering (grade of C- or better)

Or their equivalents at another school. Student records will be reviewed to determine whether a student has taken the required prerequisites and if not (s)he will be dropped from the class.

Skills Prerequisite:

To perform well in this course, you must have all the basic knowledge of the following topics:

1. Programming in Java.
2. Using Java IDE such as Eclipse.
3. Basic understanding of object-oriented programming, such as C++, C#, Java, python
4. Implementation of common data structure, such as stack, tree.
5. Pre-calculus math including trigonometric functions, Cartesian coordinates, points, lines, coordinate transformations, inequalities, and matrix operations.
6. Software engineering techniques, such as design pattern.
7. Unified Modeling Language (UML) Class diagrams, and using UML drawing software.
8. Basic computer skills with internet connection such as downloading files, using Canvas, watching video.

Required Material:

Textbook: None

Suggested Reading: The following is the list of the suggested reading. All books are available in our CSU library system. Availability are limited.

- Object-Oriented Design & Patterns, Second Edition, by Cay Horstmann; Wiley; ISBN 0-471-74487-5
- Schaum's Outlines: Computer Graphics, Second Edition, by Zhigang Xiang, and Roy Plastock; McGraw-Hill; ISBN 0-07-135781-5
- Design Patterns : Elements of Reusable Object-Oriented Software, First Edition, by Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides; Addison-Wesley; ISBN : 978612433290
- Codename One Developer Guide, Shai Almog, Steve Hannah, and Chen Fishbein; Online: <https://www.codenameone.com/developer-guide.html>

Required Equipment:

- A computer with internet connection, monitor and speakers.
- Webcam and microphone (for Online)
- Java Development Toolkit (see Module 1.0 for installation guideline)
- Eclipse SDK (see Module 1.0 for installation guideline)
- Codename One 6.0 Plugin (see Module 1.0 for installation guideline)
- JavaFX (see Module 5.0 for installation guideline)

If you do not have the required equipments, there are several computer options available for this course. You may work on any school machine in our lab, including ECS Open Labs [RVR 2011, SCL 1234, SCL 1208 (24-hour lab)], ECS Teaching Labs [ARC 1014/1015 (classroom instruction only labs)], and CSC Labs [RVR 1013/2005/2009/2013/5029]. ECS remote lab access called Sassafras via <https://stats.labs.csus.edu/maps/ECS> are also available. Instructions for the lab and remote can be found at the EVS Resource Center Canvas Course (<https://csus.instructure.com/courses/78368>). Typically, the requirement software and hardware are installed in those lab. For more information about the lab resource, you can read the [Resource](#) section in Module 0.

Course Learning Objectives

After finishing this course, students will be able to perform the objectives listed in the following table:

Table: The course learning objectives and their evaluation methods.

#	Course Learning Objectives
1	Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
2	Apply computer science theory and software development fundamentals to produce computing based solutions.
3	Develop software using object-oriented (OO) programming language
4	Outline commonly used design patterns, and develop software using them
5	Construct GUIs to control the software
6	Use basic interactive technique for application development
7	Explain Coordinate system and its transformation
8	Develop a complete application from scratch to fulfil provided requirements

Learning Activities

Each module contains an overview page with an associated lectures and quiz. Some modules contains assignment, vocabulary. And there will be a final examination. Students need to read through all the module material and then complete the activities in each module to complete this online course.

Online Lectures

Each module contains material for online lecture. Student can attend the online lectures via the Zoom meeting on the left side menu of the Canvas. Student are required to understand every materials in the lectures. The reading is for supplement in case students cannot attend the lecture.

Programming Assignment

As a programming intensive course, it is important to have programming assignment. In this course, you will have a semester-long accumulating assignment. In other words, you need to build your next assignment on top of the previous one. Typically, there will be four programming assignment. Read the [Assignment Guideline](#) for more information.

Final Exam

At the end of this course, there will be one final examination. It includes short questions like multiple-choice, fill-in-the-blank, matching. It also includes long questions like essays, programming, drawing, calculation, etc. The topic covers the entire course including the programming assignment.

Schedule (Summer 23)

Date	CSC133-01	CSC133-02
05/30 (T)		Lecture 1
05/31(W)	Lecture 1	
06/01(R)		Lecture 2
06/05(M)	Lecture 2	
06/06 (T)		Lecture 3, A1 deadline
06/07(W)	Lecture 3	
06/08(R)		Lecture 4
06/10(Sa)	A1 deadline	
06/12(M)	Lecture 4	
06/13 (T)		Lecture 5
06/14(W)	Lecture 5	A2 deadline
06/15(R)		Lecture 6
06/18(Su)	A2 deadline	
06/19(M)	Holiday	Holiday
06/20 (T)		Lecture 7
06/21(W)	Lecture 6	
06/22(R)		Lecture 8
06/23(F)		A3 deadline
06/26(M)	Lecture 7	
06/27 (T)	A3 deadline	Lecture 9
06/28(W)	Lecture 8	
06/29(R)		No Class
06/30(F)		A4 deadline
07/03(M)	Lecture 9	
07/04(T)	Holiday, A4 deadline	Holiday
07/05(W)	Final	
07/06(R)		Final

Policies

Grading

In this online course, the ratios of different activities on the final grade are as following:

1. Programming assignment (65%)
2. Final examination (35%)

At the end of the semester, the scores of each student will be curved based on the all students in the class. And then a grade of each student will be calculated based on the following table:

The grading scale used by this course

Grade	F	D-	D	D+	C-	C	C+	B-	B	B+	A-	A
Score	< 50	50-53	53-56	56-60	60-66	66-72	72-78	78-83	83-88	88-92	92-96	> 96

where C- is the minimum credit grade for CSC133 course.

To achieve a credit final grade of at least C- in this course, you **must** achieve at least passing completion (that is 50%) of **BOTH** the average of programming assignments; and the final exam. In other words, if your assignment average or final exam does not achieve 50% or more, the maximum grade you can achieve in this course is D+.

Late Policy (Spring/Fall)

For the programming assignment in the spring or fall semester, late assignments will be only accepted up until certain days past the original due date. I.e., assignments submitted **ten** days after the deadline (and beyond) will not be accepted. A penalty of 5% per day will be applied to all late work, and the maximum late penalty will be 50%. School holidays and weekends will be counted as regular days in computing the lateness of assignments. For example, if an assignment were due on a Friday and were submitted on a Monday, it would be counted as three days late. If the student achieved a 70 score for that assignment, it becomes $70 * (1 - 0.05 \times 3) = 59.5$ after the late penalty.

Late Policy (Summer)

For the programming assignment in the summer semester, late assignments will be only accepted up until certain days past the original due date. I.e., assignments submitted **five** days after the deadline (and beyond) will not be accepted. A penalty of 10% per day will be applied to all late work, and the maximum late penalty will be 40%. School holidays and weekends will be counted as regular days in computing the lateness of assignments. For example, if an assignment were due on a Friday and were submitted on a Monday, it would be counted as three days late. If the student achieved a 70 score for that assignment, it becomes $70 * (1 - 0.10 \times 3) = 49$ after the late penalty.

Attendance

There is no attendance requirement for this course. However, it is very important for you to read all the lecture materials and complete all the learning activities on time in order to learn.

Drop/Withdrawal

Please inform yourself of the Department, College, and University policies on dropping courses. Policy information is available in the Computer Science Department office, Riverside Hall 3018. General add/drop policy of CSU can be found on the [Registration](#) page of our university.

Repeat

Department policy specifies that students may not repeat a Computer Science course more than THREE times (that is, take a course for a fourth – or subsequent – time). Any student who wishes to repeat a course more than three times must submit a petition requesting permission to do so. Student records will be reviewed to determine whether a student is taking this course for a fourth (or subsequent) time. Any such student must return an approved petition to the instructor within the first two weeks of class. Any student who does not submit an approved petition will be dropped from the class. Petitions are available in the Department office (Riverside Hall 3018) and require the signature of both the Instructor and the Department Chair.

Ethics

When a student submits work to the instructor, it constitutes a contractual agreement that the work is solely that of the student. Further, submitted work carries with it an implicit agreement that the instructor may quiz the student in detail about the work. Since the class is graded on a curve, if a student attempts to raise their grade through unethical means it is in essence causing a lowering of the grades of other students in the class. Further, any student who gives such help is equally guilty of unethical behavior for the same reasons. Therefore, all students are hereby notified that this course is being taught by instructors who will pursue attempts at cheating, since students who are cheating are cheating on you. The minimum penalty for even a single incident of cheating in this course is **automatic failure of the course**; additional more severe penalties may also be applied. Note that cheating is grounds for dismissal from the University. Please refer to the instructors' policy on academic integrity entitled "[Ethics in Computer Science Classes](#)" to the Computer Science Department's document entitled "[Policy on Academic Integrity](#)" and to the University's document entitled "[Policy Manual section on Academic Honesty](#)". It is the responsibility of each student to be familiar with, and to comply with, the policies stated in these documents.

Confidentiality

The materials in this course is designed by your professors and other instructors of other section. We own the copyright and all right reserved. You are allowed to share the materials to other students in California State University, Sacramento (CSUS) with proper credit. However, you are not allowed to share to the other people outside CSUS. For instance, uploading material to Github and open to general public is violating the policy and not allowed. For the programming assignment, the program design copyright is owned by your professor. After enrolling in this course, you will gain the permission to implement the program and you agree that you will never publish your assignment code to the public without the consent from the copyright owner. Violating this policy can make you fail in this course.

Change of document

Instructor has the right to change this document anytime without further notice to the students. It is the student's responsibility to check Canvas regularly for corrections or updates. Instructor will announce important changes in the [Announcement section](#) if necessary.

Questions?

If you have any general questions or concerns about the course, please send email to your instructor.

Click "Next" to see what kind of supporting resource can you get from our University.



[◀ Previous](#)

[Next ▶](#)