

Course Syllabus for CIS554, Fall 2020

- All lectures will be live-streamed and recorded using zoom.
- Guidelines for attending classes in person: (1) Wear face mask (2) Stay 6 feet apart (3) Switch to “on line” if not feeling well.
- All office hours for instructor and TA will be “on line”.

COURSE INFORMATION

Title: Object Oriented Programming in C++

Instructor: C-Y (Roger) Chen **Email:** crchen@syr.edu

Office Location: CST 4-281

Office Hours: 2:20PM-3:10PM Tuesday & Thursday

TA: Abel Armede (agarmede@syr.edu); Office Hours: 10-12AM Wednesdays & Fridays; (TA hours might change according to HW due dates.)

COURSE DESCRIPTION (from Catalog)

Survey of basic C constructs. Data abstraction, classes, derived classes, types, structures and template. Access control, information hiding, multiple inheritance. Formatting stream I/O, libraries, interfaces, modular system Organization. Substantial programming assignments.

Goal: Proficiency in C++ programming and Reasoning in C++ programming styles

COURSE OBJECTIVES

The students will...

- Review the classical software design concepts of functional decomposition.
- Subsequently learn object-oriented design concepts.
- Be able to design from a functional decomposition perspective.
- Be able to develop similar object oriented designs in C++.
- Be able to compare and contrast different design methodologies and languages.

READING MATERIAL

Will not follow any textbooks:

(Required reading: Chapters 1-10 during the first two weeks) C++: A Beginner's Guide, 2nd Edition, Author: Herbert Schildt

Primary sources of lectures: Material prepared by the instructor

Reference: The C++ Programming Language (hardcover) (4th Edition), 2013 by Bjarne Stroustrup

GRADING & ASSESSMET PROCESS

Homework: 6-8 programming assignments (about **50%** of grade)
Free Download: Microsoft Visual Studio 2019, Community Edition;
<https://visualstudio.microsoft.com/downloads/>

Late Penalty: first day **30%**, extra **10%** each subsequent day.
Homework will always be due at 11:59PM. A grace period of 30 minutes will be given. Any submission by 12:30AM will not be given late penalty. Submission at 12:31AM will be given 30% penalty.

Midterm Exam: about **16%** of grade

Final Exam: about **34%** of grade

Class Attendance: about **-10 - 0%** of grade

Class Attendance and Participation: class attendance is mandatory.
You are expected to participate in class activities and discussions. **Class attendance will be taken. Up to 2 absences will be excused.**

REQUIRED KNOWLEDGE, ABILITIES, AND SKILLS

Pre-Requisite: ESC 102 – Introduction to Computing (or other equivalent programming knowledge)

Recall: You should be able to...

- ☐ Define the Boolean operations NOT, AND, and OR.
- ☐ Define the precedence of mathematical operators.

Comprehension: You should be able to...

- ☐ When given a Boolean expression, express it using truth tables.
- ☐ When given a high-level programming statement, explain its meaning in English.

□ When given an integer number in decimal, binary, or hexadecimal representation, translate it to another base.

□ Add and subtract numbers represented in decimal, binary or hexadecimal.

□ When given a number, form its two's complement.

Synthesis: You should be able to...

□ When given a problem description, develop a C++ programming solution.

TOPICS COVERED

TENTATIVE TOPICS AND SCHEDULE

Introduction to C++, arrays, pointers, dynamic memory allocation

Linked lists operations, doubly linked lists

Operator overloading

Standard Template Library (STL)– Array, List, Vector, Map, Stack, Queue

Operator overloading, recursions

Copy constructor, destructor, deep and shallow copying

L-value, R-value, move constructor, overloaded operator=, Initializer_list, Template

Initializer_list, Templates, exception handlers

Inheritance

Functors, Lambda expressions, function pointer, smart pointers

Smart pointers, hash table

Functor, comparator class, static_cast, dynamic_cast, const_cast

Virtual class

Last day of class: Nov. 24 (Tuesday)

Midterm Exam: 10:30am – 12:00pm, October 11 (Sunday)

Final Exam: December 5 (Saturday), 9-11:45AM.

ACQUIRED KNOWLEDGE, ABILITIES, AND SKILLS

Analysis: You will be able to...

□ When given a design problem, analyze the classes required to determine if inheritance can be efficiently utilized.

Synthesis: You will be able to...

- Develop code for object-oriented designs in C++.
- Implement constructors and destructors.
- Design classes to implement encapsulation in C++.
- Formulate strategies for data hiding using access rights and access methods.
- Develop tests to verify correct operation of designs.
- When given a design problem, develop a class/object model, dynamic model, and a functional model.

Evaluation: You will be able to...

- Compare and contrast the different design methodologies and languages.
- Evaluate an existing C++ implementation to determine what object-oriented design models are the bases of the program.
- Evaluate designs for reuse.

Syracuse University Academic Integrity Policy Will be Enforced:
http://class.syr.edu/wp-content/uploads/2017/05/Academic-Integrity-Policy_final.pdf

Special Honor Policy Reminder for CIS 554

Plagiarism will be severely penalized. All assignments will be thoroughly checked for possible plagiarism.

As stated in the course syllabus, every student is expected to behave ethically: do not cheat, plagiarize, or commit fraud.

Fraud includes manipulating the simulation results to make it appear that a design functions correctly when it does not; plagiarism includes using someone else's work without proper credit. The following guidelines further detail these definitions:

1. If you happen to find a solution to a problem in a written source (e.g., in a textbook or on the web) and use the solution, you should state the original source. It is unethical and plagiarism to use someone else's work without proper credit.
2. If you get help from someone (either a classmate or someone else) on a problem, you should state that (and give their name). Again, it is unethical to use someone else's ideas without giving proper credit. In both of the above cases, it is cheating and fraud to pass off something as being your work when it is not.
3. In this course, it is legitimate to discuss problems with each other at a conceptual level: for example, it's okay to figure out how to break a design problem up into smaller, easier-to-design modules, or to discuss general approaches to solving a problem. However, the final design solution must be your own: it is not legitimate to share any details, layouts, or other written solutions, or to discuss how to implement any design modules with anyone.
4. If you are unsure whether a certain action constitutes cheating, fraud, or plagiarism, assume that it does: you may ask me for clarification at any time.