

Introduction to Programming Syllabus

1. Basic Course Information

Course Name:	Introduction to Programming		
Course Level:	Undergraduate	Credit/Contact Hour:	4/64
Major	All	Teaching Language:	English
Prerequisite:	None		
School/Institute:	SIST	Course Code:	CS100

Notes: Course level includes undergraduate, graduate, or undergraduate/graduate.

2. Course Introduction

Programming is an important fundamental skill across many engineering disciplines. The present course provides an introduction to programming for starting undergraduate students. The course will be based on C and the object-oriented C++ programming languages, thus equipping attendees with powerful low-level language knowledge for the efficient implementation of resource-hungry programs. The course runs over 16 weeks and puts a strong emphasis on practical abilities. This is underlined by the presence of weekly recitations where every student will be regularly advised as part of a group of about 20-30 students. Recitation sessions will review class material and introduce further material in a more interactive format that involves practice coding sessions. The grading structure also puts a strong emphasis on practical experience with the majority of the score being made up by the homework grades. A late midterm exam is kept as a more traditional grading measure.

3. Learning Goal

Cognitive competence:

- Learn C and C++ features including later standards

- Understand various ways of programming such as sequential, object-oriented, recursive, and functional programming
- Learn about polymorphism and the standard template library
- Learn about concurrency
- Learn about modern C++ standards
- Learn about cross-compilation, profiling, debugging, and interfacing

Comprehensive qualities:

- Gain skills in problem solving and modular thinking
- Valuable practical engineering skills for the efficient solution of real-world problems

4. Instructional Pedagogy

- 4 hours lecture per week, taught in English, during 16 weeks
- 2 hours recitation per week, in Chinese, during 15 weeks (starting from week 2)
- 8 homework assignments, 4 in the beginning which are of slightly smaller volume, and 4 larger ones towards the end which will be more challenging
- 1 midterm exam
- 2 quizzes to probe class attendance
- Online forum (piazza) to answer questions
- Office hours offered by both Prof. and TAs

5. Course Structure

(You might choose one of the two course structures listed below. Please list practical training if any.)

Course Structure by Chapter

Chapter	Teaching Contents	Week	Contact Hours	Teaching Modes
Introduction	C syntax, development environment	1	4	Lecture, forum, recitation,

				homework, quiz, mid-term
Basics	Flow Control, types, arrays, operations	2	4	Lecture, forum, recitation, homework, quiz, mid-term
Code structuring	Procedural abstraction, functions	3	4	Lecture, forum, recitation, homework, quiz, mid-term
Pointers	Pointers, Pointer arithmetic	4	4	Lecture, forum, recitation, homework, quiz, mid-term
Algorithms	Sorting	5	4	Lecture, forum, recitation, homework, quiz, mid-term
I/O	Console I/O, debugging	6	4	Lecture, forum, recitation, homework, quiz, mid-term
Recursion	Recursive programming	7	4	Lecture, forum, recitation, homework, quiz, mid-term

OO Programming	Object-Oriented language, Composition/Aggregation/ Inheritance, virtualization	8	4	Lecture, forum, recitation, homework, quiz, mid-term
Polymorphism	Polymorphism, templates	9	4	Lecture, forum, recitation, homework, quiz, mid-term
Data structures	STL, vector, list, map	10	4	Lecture, forum, recitation, homework, quiz, mid-term
Concurrency	Processes, threads, hyper- threading, thread synchronization	11	4	Lecture, forum, recitation, homework, quiz, mid-term
Standards	Coding standards, Coding patterns	12	4	Lecture, forum, recitation, homework, quiz, mid-term
Code management	CMake, Documentation, Profiling, Debugging	13	4	Lecture, forum, recitation, homework, quiz, mid-term
New standards	R-value references, C++11, C++14, C++17	14	4	Lecture, forum, recitation, homework, quiz, mid-term

Math in C++	Eigen, Ceres	15	4	Lecture, forum, recitation, homework, quiz, mid-term
Interfacing	Interfacing with Matlab and Python	16	4	Lecture, forum, recitation, homework, quiz, mid-term

6. Grading Policy

The evaluation will be as follows:

- 4 times 7 points for the shorter homeworks in the beginning of the course
- 4 times 11 points for the longer homeworks towards the end of the course
- 22 points for the midterm exam
- Some points for probing class attendance etc.

7. Textbook & Recommended Reading

(1) Textbook

- The C Programming Language (2nd Edition), by Brian W. Kernighan and Dennis M. Ritchie, Prentice Hall, 2014
- The C++ Programming Language, 4th Edition, Bjarne Stroustrup, Pearson, 2013
- Effective C++, 3rd Edition, Scott Meyers, Addison-Wesley, 2005
- C++ Primer, Josée Lajoie and Stanley B. Lippman

(2) Recommended Reading

- <https://en.cppreference.com/w/>
- <https://www.clpusplus.com/reference>
- <https://stackoverflow.com/>

8. Academic Integrity

This course highly values academic integrity. Behaviors such as plagiarism and cheating are strictly prohibited. **You may ...**

- ... discuss with your peers about course material, and go over it together.
- ... discuss generally about a/the programming language, some features, or abstract lines of code. As long as it is not directly related to any homework, but formulated in a general, abstract way, such discussion is acceptable.
- ... do joint practice coding, as long as the code in question is not directly related to any homework, but framed in a general abstract way.
- ... share test cases with each other.
- ... help each other with setting up the development/build environment etc.
- ... discuss the homework instructions to clarify what they mean (without discussing the solution!!)
- ... search online to help identify solutions to your problem.

You may not ...

- ... **read, possess, copy or submit the homework solution code of anyone else** (including people outside this course or university)!
- ... **receive direct help on homework from someone else** (i.e. a direct communication of some lines of code, no matter if it is visual, verbal, or written)!
- ... **give direct help on homework to someone else**. Helping one of your peers by letting him read your code or communicating even just part of the solution in written or in verbal form will have equal consequences.
- ... gain access to another one's homework related OJ/computer/online code repository account, no matter if with or without permission.
- ... give your account access to another student. It is your responsibility to keep your account safe, always log out (even if it is only for short term), and choose a safe password. Do not just share access to your computer with other students without prior lock-out and disabling of automatic login functionality. Do not just leave your computer on without a lock even if it is just for the sake of a 5-minute break.
- ... work in teams. You may meet to discuss generally about the material, but any work on the homeworks is to be done individually and in privacy. Remember, you may not allow anyone to even just read your source code. Find a quiet corner where nobody can see your screen when you work on your homework.

- ... just copy-paste a solution from an online webpage. It will most likely not do exactly what you want, and it increases the likelihood that your code will have strong similarities with someone else's code. You need to understand any code you submit, which likely means that you will have to fully understand and adapt any code you find online to your own needs.

With the Internet, "copy", "paste", and "share" are easy operations. Don't think that it is easy to hide and that we will not find you, we have just as easy to use, fully automatic and intelligent tools that will identify any potential cases of plagiarism. And do not think that being the original author will make any difference. Sharing an original solution with others is just as unethical as using someone else's work.