

Introduction to Computer Programming II – CSI 32, Sec. 52112

Professor: Dr. Luis Fernández

Class times and room: Tu, Th, 10:00 to 11:50 at CP 320.

Course page: <http://fsw01.bcc.cuny.edu/luis.fernandez01>

Office & Tel.: CP 301. (718) 289-5100, Ext. 3209.

Office hours: Tu 13–14, Th 9–10, or by appointment.

e-mail: luis.fernandez01@bcc.cuny.edu

Overview of the course.

This course is a continuation and enhancement of CSI 31. The goal is to learn about computer structures using the C++ programming language. C++ is one of the oldest languages that are still widely used. Most languages used nowadays—java, perl, javascript, python, php—are based on or heavily influenced by C++. It offers a lower level view of computing than python or java, for example, in the sense that one can get closer to the ‘guts’ of the computer. In addition, it offers faster performance than most other languages. At the end it is expected that the student will be able to:

- Program with the object-oriented concepts classes, objects, data members, member functions and create classes.
- Use pointers and built-in arrays.
- Access class members and learn the order of constructor and destructor calls.
- Use operator overloading.
- Understand polymorphism, inheritance, use constructors and destructors in inheritance hierarchies.
- Use C++ object-oriented stream input/output operations.
- Build C++ programs that create, update and process data files.
- Understand Exception Handling, use try, catch and throw.
- Program with Linked lists.
- Implement various sorting algorithms.

Some resources:

- **Classes:** Attendance is essential to succeed in the class. In class you will have time to learn new material, practice, and ask questions.
- **Meetings with the instructor:** If you need help, or for any other matters concerning the course, you can talk to me during office hours or at other time we arrange.
- **Email:** If you have questions while doing homework and need help quickly, please email me anytime (address above).
- **Finding examples and explanations online:** The internet is the programmer’s best friend. If you have a question or you do not understand an error, copy it into your browser and you will find many answers.

Textbook:

- *C++ How To Program, 10th Edition*, by P. Deitel and H. Deitel, 2017. You need to have to book because you will be asked to do exercises from it. You can find it online new or used at a good price.

Highly recommended book:

- *Programming: Principles and Practice Using C++*, by Bjarne Stroustrup. The author was the *creator* of C++, and the book is a great resource to understand basic concepts and learn programming from a pro.

Student’s responsibilities

- To use the **resources** available (some are above) to attain the main goal: to learn.
- To work on many **exercises**, as it is impossible to learn programming without doing so.
- To **ask** questions during class or tutorials about anything that has not been understood.

Instructor’s responsibilities

- To act as *facilitator* of the learning process of the students, and to assist with any question that students may have about the material.
- To give tests and exams of appropriate difficulty. To grade tests and exams promptly and explain the students the meaning of their grades.

Some Rules

- Tests will not be repeated. The only exception, in some situations, is if the instructor receives notice of the absence (via e-mail or telephone) **on or before the day of the test or quiz**.

Exams and homeworks:

- **Weekly homework** will be assigned and submitted via Dropbox. It counts **20%**.
- A **final project**, worth **25%** of the final grade.
- Two **in class tests**, worth **15%** of the final grade each, totalling **30%**.
- The **final exam** will count **25%** of the final grade.

Notes about programming and homework:

- Programming is only learned by **practice**. It does not get easier, but you get smarter and more experienced.
- **Every** program will have errors when you first compile it. Learning how to fix errors is one of the most important skills you need to learn. Try different things: read the error, find the lines where the error happened (always look at the lines before also). If you do not see where is the error, comment out some lines to narrow your search. Use a debugger if necessary. Do not give up!!
- If something does not work, modify it and see what happens, even if you do not understand what is going on. Using **trial and error** you can write very complicated programs without even knowing the language.
- Only programs that **run** will be accepted in assignments. Even if it does not do what it should, it must run.
- Submission of homework must be done into your folder in Dropbox.

Class plan and assigned exercises.

Date	Section From Text	Assignment From Text	To Submit
Tu 1/28	Ch. 1. Intro to computers and C++. Must read .	1.1–1.3	
	Ch. 2. Intro to C++, Input/Output and Operators.	2.1–2.6, 2.16, 2.18, 2.19, 2.23, 2.24, 2.25, 2.28, 2.29.	2.19, 2.18.
	Ch. 2. Intro to C++, Input/Output and Operators.	2.1–2.6, 2.16, 2.18, 2.19, 2.23, 2.24, 2.25, 2.28, 2.29.	2.19, 2.18.
Th 1/30	Ch. 4. Algorithm Development, Control Statements, 1.	4.1–4.10, 4.13, 4.15, 4.17, 4.19, 4.26, 4.27, 4.29, 4.35	4.26, 4.35 a, b
Tu 2/4	Ch. 4. Algorithm Development, Control Statements, 1.	4.1–4.10, 4.13, 4.15, 4.17, 4.19, 4.26, 4.27, 4.29, 4.35	4.26, 4.35 a, b
Th 2/6	Ch. 5. Algorithm Development, Control Statements, 2.	5.1–5.4, 5.12, 5.14, 5.15, 5.16, 5.19, 5.23, 5.24	5.15, 5.19, 5.24
Tu 2/11	Ch. 5. Algorithm Development, Control Statements, 2.	5.1–5.4, 5.12, 5.14, 5.15, 5.16, 5.19, 5.23, 5.24	5.15, 5.19, 5.24
Th 2/13	Ch. 5. Algorithm Development, Control Statements, 2.	5.1–5.4, 5.12, 5.14, 5.15, 5.16, 5.19, 5.23, 5.24	5.15, 5.19, 5.24
Tu 2/18	Ch. 6. Functions and recursion.	6.1–6.10, 6.12, 6.13, 6.14, 6.16a, b, 6.18	6.18
Th 2/20	Ch. 6. Functions and recursion.	6.20, 6.21, 6.22, 6.23, 6.24	6.23, 6.24
Tu 2/25	Ch. 6. Functions and recursion.	6.26, 6.28, 6.29, 6.31, 6.33, 6.38, 6.41	6.29, 6.31, 6.38
Th 2/27	TEST 1: Ch. 1 to 6.		
Tu 3/3	Ch. 7. Arrays and vectors; catching exceptions.	7.1–7.5, 7.11, 7.13, 7.14, 7.16	7.16
Th 3/5	Ch. 7. Arrays and vectors; catching exceptions.	7.22 (in groups)	7.22 (in groups)
Tu 3/10	Ch. 7. Arrays and vectors; catching exceptions.	7.22 (in groups)	7.22 (in groups)
Th 3/12	CLASSES SUSPENDED due to pandemic		
Tu 3/17	CLASSES SUSPENDED due to pandemic		
Th 3/19	Ch. 8. Pointers.	8.1–8.6, 8.7, 8.12 (in groups)	8.12 (in groups)
Tu 3/24	Ch. 8. Pointers.	8.15 (in groups)	8.15 (in groups)
Th 3/26	Ch. 8. Pointers.	8.15 (in groups)	8.15 (in groups)
Tu 3/31	Ch. 3. Intro to classes, objects and member functions.	3.1–3.4, 3.10, 3.11, 3.12	3.12
Th 4/2	Ch. 3. Intro to classes, objects and member functions.	3.1–3.4, 3.10, 3.11, 3.12	3.12
Tu 4/7	NO CLASS - WEDNESDAY SCHEDULE		
Th 4/9	NO CLASS - SPRING BREAK		
Tu 4/14	Ch. 9. Classes: A deeper look.	9.1, 9.2, 9.5, 9.6, 9.7, 9.8, 9.11	9.6
Th 4/16	Ch. 9. Classes: A deeper look.	9.14, 9.15, 9.20, 9.21, 9.23	9.23
Tu 4/21	Ch. 10. Operator overloading.	10.1–10.5, 10.7, 10.8, 10.9, 10.10	10.10
Th 4/23	TEST 2: Ch. 1 to 10		
Tu 4/28	Ch. 11. Object-oriented programming: Inheritance.	11.1, 11.2, 11.6, 11.9 (in groups)	11.9 (in groups)
Th 4/30	Ch. 11. Object-oriented programming: Inheritance.	11.1, 11.2, 11.6, 11.9 (in groups)	11.9 (in groups)
Tu 5/5	Ch. 12. Object-oriented programming: Polymorphism.	12.1, 12.2, 12.11	
Th 5/7	Ch. 12. Object-oriented programming: Polymorphism.	12.12 (in groups)	12.12 (in groups)
Tu 5/12	Ch. 13. Stream input/output: A deeper look.	13.1–13.5, 13.6, 13.7, 13.8, 13.10	13.6, 13, 13.7
Th 5/14	Ch. 14. File processing.	14.1–14.4, 14.6, 14.7, 14.8	14.8

Academic Integrity:

Academic dishonesty (such as plagiarism and cheating) is prohibited at Bronx Community College and is punishable by penalties, including failing grades, dismissal and expulsion. For additional information and the full policy on Academic Integrity, please consult the BCC College Catalog.

Accommodations/Disabilities:

Bronx Community College respects and welcomes students of all backgrounds and abilities. In the event you encounter any barrier(s) to full participation in this course due to the impact of a disability, please contact the disAbility Services Office as soon as possible this semester. The disAbility Services specialists will meet with you to discuss the barriers you are experiencing and explain the eligibility process for establishing academic accommodations for this course. You can reach the disAbility Services Office at: disability.services@bcc.cuny.edu, Loew Hall, Room 211, (718) 289-5874.