CSCI271(04) - Introduction to Computing and Programming (C++)

COURSE SYLLABUS
Spring 2025

I am committed to creating a course that is inclusive in its design. Students who need an accommodation due to a disability are encouraged to contact the Office of Accessibility Services (OAS) within the first week of class or as soon as possible thereafter. Requests for Accommodations can only be approved by the OAS. Students with disabilities are entitled to confidentiality over disability-related status or details. Students are not required to disclose their specific disability to their instructors or anyone else. All OAS approved accommodations will be honored and implemented in my course.

Contact OAS, New Building, room L.66
Phone: (212) 237-8031
Email: accessibilityservices@jjay.cuny.edu.

Course Details

Tuesdays & Thursdays
8:00AM – 9:15AM

Class Venue: New Building (NB) L2.79

3 credits

Instructor : Temidayo Akinyemi

Email: takinyemi@jjay.cuny.edu

Office Hours: Mondays 10AM–11AM via Zoom

Text and Reference Material



C++ How to Program (10th Edition) By Paul Deitel & Harvey Deitel ISBN-13: 978-0-13-444823-7



C++ Reference: http://en.cppreference.com/w/cpp



C++ FAQ: https://isocpp.org/faq

Course Requirements



Prerequisites: ENG 101, and MAT 105 or the equivalent.



This course satisfies the Flexible Core: Scientific World area of the Gen Ed Program and is a Core Computer Science course in the Computer Science and Information Security, and Applied Mathematics majors.



This course is also a prerequisite for CSCI 272 (Object-oriented Programming)

Course Objectives



Identify the major logical units of a computer, understand the difference between machine, assembly and high-level computer languages and explain the process that source code goes through to become an executable program



Demonstrate the ability to design programs to take input from a keyboard and output text to a computer terminal window.



Understand operator precedence and demonstrate the ability to use operators to perform arithmetic calculations and assign values to variables.



Demonstrate the ability to build simple conditions, as well as more complex conditions using boolean logical operators, in selection control structures, executing the appropriate code according to the result of the condition evaluation.

Course Objectives Cont'd



Explain the difference between sequence, selection and repetition control structures and demonstrate the ability to use each to implement algorithms.



Understand the rationale behind dividing program tasks into separate functions, the operation of the function call stack and the difference between passing arguments by value and by reference and demonstrate the ability to construct and use functions.



Demonstrate the ability to declare and use single and multi-dimensional arrays, within a function or passed between functions.



Explain what a pointer is, how pointers relate to arrays, and demonstrate the use of pointers and pointer operators.

Course Policies

- This course will use **Brightspace** for announcements, lecture notes, code samples, videos and some assignments.
- All assignments (labs, homeworks and others) must be submitted on time. Late submissions will not be accepted. Labs are in-person assignments, so if a student is not in class when the lab is given, they will receive no points.
- Projects may be accepted up to a week late if instructor is contacted prior to submission date. Late projects may attract 20% loss
- This course will use **https://repl.it** for programming exercises. repl.it is an online coding environment for C++ and other programming languages. (may also use a c++ compiler at **onlinegdb.com**)
- You're welcome to use **your own laptop** instead of the lab computer to work on code. During tests however, no electronic devices, including laptops, smartphones and smartwatches, may be used.

Use of AI

- This course welcomes the use of AI for learning programming concepts.
- AI will be embedded in various ways in the teaching and learning process throughout this course. However, only instructor guided and approved forms of use will be accepted.
- The use of AI to obtain answers to labs, tests, exams is strictly forbidden. Any student found in violation will be referred to the campus Academic Integrity Office.

Resources for Student Success

- Learning to program takes practice! You are encouraged to work on your own programming challenges outside of course assignments.
- If your program won't compile and you can't figure out why, you are welcome to post in the Help discussion forum on Blackboard. Other students undoubtably have the same question. The instructor will monitor the Help forum regularly and students may also help each other out

- The Math and Science Resource Center (MSRC) offers tutoring on an appointment basis. See this link for more information: http://www.jjay.cuny.edu/mathematicsscience-resource-center



Grading

Labs & Homeworks	20%
Unit Tests (3)	40%
Projects	20%
Final exam	20%
TOTAL	100%

Overall course grading will follow the John Jay College standard in the *Undergraduate Bulletin*.

A 93.0-100.0	**B- 80.0- 82.9**	**D+ 67.1- 69.9**
A- 90.0- 92.9	**C+ 77.1- 79.9**	**D 63.0- 67.0**
B+ 87.1- 89.9	**C 73.0- 77.0**	**D- 60.0- 62.9**
B 83.0- 87.0	**C- 70.0- 72.9**	**F Below 60.0**

Class Schedule

This class will not be held on the following days according to the John Jay Academic Calendar:

- Tuesday, February 18, 2025 = Classes follow a Monday scheduled
- Thursday, March 6, 2025 = Classes follow a Wednesday schedule
- April 15 & 17, 2025 = Spring Break

Tentative Schedule of Topics (subject to change)

Unit 1	**Introduction (Jan. 25 - Feb. 15)** • "Hello World" • Computers, Hardware and Software • Data Hierarchy • Types of Languages • C++ Development Environment • Basic Output • Variable Declaration and Input • Arithmetic Operators & Operator Precedence	Chapter 1 Chapter 2, up to section 2.6
Unit 2	**Decision Making (Feb. 20 - Mar. 14)** • equality and relational operators • if statements • if / else statements • switch statements • boolean logical operators	Chapter 2, section 2.7 Chapter 4, sections 4.2 – 4.6 Chapter 5, section 5.9, 5.11
Unit 3	**Repetition (Mar. 19 - Apr. 4)** • sequential, selection and repetition control structures • while loops • do … while loops • for loops • parallel programming – loops	Chapter 4, sections 4.8 – 4.13 Chapter 5, sections 5.1 – 5.8, 5.10

	• counter-controlled repetition vs. sentinel- controlled repetition • nested control structures • assignment operators • increment/decrement operators • break and continue statements	
Unit 4	**Functions (Apr. 9 - Apr. 25)** • function definitions and prototypes • C++ standard library functions (math) • random number generation • scope rules • function call stack • pass by value vs. pass by reference • default arguments • unary scope operator • function overloading • introduction to recursion	Chapter 6
Unit 5	**Arrays and Pointers (Apr. 30 - May 15)** • declaring and using one-dimensional arrays • multi-dimensional arrays • searching arrays • using arrays with functions • pointer declaration and initialization • address and dereference operators • using pointers with functions • pointers and arrays	Chapter 7, sections 7.2 – 7.5 Chapter 8
	Departmental Final Exam: Tuesday, May 20, 2025	

Academic Integrity: Definitions and Examples of Academic Dishonesty

- **Cheating**
  - Copying from another student during an examination or allowing another to copy your work
  - Unauthorized collaboration on a take-home assignment or examination
  - Taking an examination for another student, or asking or allowing another student to take an examination for you
  - Allowing others to research and write assigned papers or do assigned projects, including use of commercial term paper services
  - Giving assistance to acts of academic misconduct/dishonesty
  - Fabricating data (all or in part)
  - Submitting someone else's work as your own

- **Plagiarism**
  - Copying another person's actual words without the use of quotation marks and footnotes attributing the words to their source
  - Presenting another person's ideas or theories in your own words without acknowledging the source
  - Using information that is not common knowledge without acknowledging the sources