New Jersey Institute of Technology University Heights Newark, NJ 07102
http://njit.edu

CS 116 Introduction to Computer Science II in C++ – Fall 2025

Part I: Course and Instructor Information

Semester	Fall 2025
Course	CS 116
Instructor	Prabhat Vaish
Course Meeting	Please see your section details
Office Hours	

Part II: Course Description

1. **Course description:**

   This is a course in advanced programming, with an emphasis on data structures. It focuses on problem solving and use of efficient algorithms to solve commonly recurring problems in computer science. In addition to teaching the fundamental notions of structuring a programming solution, and how to write modular and elegant code, the course emphasizes finding most efficient ways to organize data, write efficient code, and learn effective programming techniques. We believe that this aspect of programming is the most important in practice, the least well-understood by our students, and the one that is hardest to acquire in standard industrial settings.

   A study of advanced programming topics with logical structures of data, their physical representation, design, and analysis of computer algorithms operating on the structures, and techniques for program development and debugging. Course covers program specifications, correctness and efficiency, data abstraction, basic aspects of simple data structures, internal searching and sorting, recursion and string processing. Algorithmic analysis is also discussed.

   This is a second course in programming, and will assume that students are well-versed with C++ programming and have written several programs, both in and outside the class.

   For the latest course information go to https://njit.instructure.com/courses/13334
   The information below should help you plan and organize your preparation during the semester.

2. Prerequisite courses and knowledge:

- Prerequisite course: CS 115
- Required background:
  - The students are required to have experience programming with an object-oriented programming language (e.g., Java, C++, Python). At a minimum, you should be familiar with the following concepts:
    - data types
      - declarations
      - limits/ranges
      - Boolean, numerical, string types
    - variables
      - assignment
      - initialization
      - primitive vs. reference types (not all languages!)
      - scoping and lifetime
    - primitive operators and semantics
    - methods/functions
      - definition
      - invocation
      - parameters & return values
    - classes and objects
      - definition
      - instantiation
      - class vs. instance variables/methods
    - control flow constructs
      - branches (if-then-else, switch)
      - looping (for, while, do-while)
    - simple input/output (I/O)
    - simple arrays and array iteration

  - The students should be able to:
    - Analyze and explain the behavior of simple programs involving the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, functions, pointers
    - Write a program that uses each of the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, functions, pointers
    - Break a problem into logical pieces that can be solved (programmed) independently.

3. Outcomes expected upon the completion of the course:

a. Use a suitable programming language, and development environment, to implement, test, and debug algorithms for solving real-life problems.
b. Write programs that use each of the following data structures (and describe how they are represented in memory): strings, arrays, structures (stacks, queues, trees, and graphs)
c. Explain the basics of the concept of recursion. Write, test, and debug simple recursive functions and procedures.
d. Explain and apply object-oriented design and testing involving the following concepts: data abstraction, encapsulation, information hiding, sub-classing, inheritance, templates
e. Solve problems by creating and using sequential search, binary search, and quadratic sorting algorithms (selection, insertion)
f. Determine the time complexity of simple algorithms.

4. **Assessment throughout the course:**
   - Quizzes and assignments - content, understanding of methods discussed in class and their effective use or application to the assignment.
   - Class participation – open contribution to the discussions and exercises, sharing, collaboration
   - Mid-term and Final exams – understanding of the course material and demonstrated effective application of the acquired knowledge and skills to solving practical problems

5. **Required & Recommended texts:**
   - ***Lecture Notes***
     Lecture notes are the basic course material for this class. The notes are made available on Canvas every week.

   - ***Text Book***
     - [**DSA**] Data Structures and Algorithms in C++, Goodrich, Tamassia, Mount, Second Edition, John Wiley & Sons, Inc.
     - [**DAC**] Data Structures and Algorithm Analysis in C++ , Mark Allen Weiss, Pearson, Hoboken, 1st Edition, 2020. https://www.amazon.com/Engineering-Software-Products-Ian-Sommerville/dp/013521064X
     - [**ECP**] Effective C++, Third Edition, Scott Meyers https://drive.google.com/open?id=19dqWms9qUQ2Bi8JfsBhC5Q6zDeJd0s5o
     - [**OADD**] Objects, Abstraction, Data Structures and Design: Using C++, Elliot B. Koffman, Paul A. T. Wolfgang, ISBN: 978-1-118-31313-8

   - ***Articles and Discussion Supporting Materials***
     For the list of readings check the Course Outline available on Canvas as well as on Discussions forum.

   - ***Books Recommended for Extra Reading***
     - "Think like a Programmer: An Introduction to Creative Problem Solving", V. Anton Spraul, 2012.
     - "Code Simplicity: The Fundamental of Software", Max Kanat-Alexander, 2012.
     - "Design Patterns / Elements of Reusable Object Oriented Software," Erich Gamma, Richard helm, Ralph Johnson, and Vlissides (known as the "Gang of 4" of "GOF"), 1994.

6. **Required software/hardware:**
   Free and open software; NJIT supported tools and hosting environments.

7. **Other Web resources:**
   See Class information on Canvas (https://njit.instructure.com/courses/13334)

Part III: Mapping Learning Outcomes to Course Assessment

Course Learning Outcome	**Measure (i.e. exam, assignment, quiz or coding)**
Use a suitable programming language, and development environment, to implement, test, and debug algorithms for solving real-life problems.	In class and online discussions; Quiz and Homework assignments
Write programs that use each of the following data structures (and describe how they are represented in memory): strings, arrays, structures (stacks, queues, trees, and graphs)	Assignments, Quiz
Explain the basics of the concept of recursion. Write, test, and debug simple recursive functions and procedures	Assignments, Quiz
Explain and apply object-oriented design and testing involving the following concepts: data abstraction, encapsulation, information hiding, sub-classing, inheritance, templates	In class and online discussions; term project
Solve problems by creating and using sequential search, binary search, and quadratic sorting algorithms (selection, insertion)	In class and online discussions; mid-term and final exam
Determine the time complexity of simple algorithms.	In class and online discussions; Homework assignments

Part IV: Course Outline (Note: this course outline is preliminary and subject to change)

Week	Lecture/Activity/Discussion	Reading (preliminary) *Check Canvas for additional reading in every module.*
Week 1 Sep 1	Course logistics and introduction Course – topics, objectives Review of C++, Arrays	Textbook: [OADD] CH P, [DSA] CH 1
Week 2 Sep 8	Problem 1 - Student Directory (OADD pg 130) HW 1 (available on 09/08) Problem-solving, OOP, Software Design C++ classes, ADT	Textbook: [OADD] CH 1, [DSA] CH 2 General Problem-Solving Techniques
Week 3 Sep 15	Quiz 1, HW 1 due (09/17) Program Correctness & Efficiency Stacks	Textbook: [OADD] CH 2, CH 5, CH 6, [DSA] CH 5
Week 4 Sep 22	Problem 2 - Finding Palindromes (OADD pg 343) HW 2 (available on 09/22) Algorithm Analysis Inheritance, Class Hierarchies	Textbook: [OADD] CH 3 [DSA] CH 4
Week 5 Sep 29	Quiz 2, HW 2 due (10/01), Linked Lists Templates, STL, Containers	Textbook: [OADD] CH 4, [DSA] CH 3
Week 6 Oct 6	Problem 3 - Customer Service Queue (OADD pg 390) HW 3 (available on 10/06) Change, Defect & Design Queues	Textbook: [OADD] CH 6 A beginner's guide to testing. Philip Johnson
Week 7 Oct 13	Quiz 3, HW 3 due (10/15) Trees, Binary trees	Textbook: [OADD] CH 8, [DSA] CH 7
Week 8 Oct 20	Review Mid Term exam	
Week 9 Oct 27	Problem 4 - Index for a Term Paper (OADD pg 507) HW 4 (available on 10/27) Hash table, Priority Queues	Textbook: [OADD] CH 9, [DSA] CH 8
Week 10 Nov 3	Quiz 4, HW 4 due (11/05) RBL Trees, Heaps	Textbook: [OADD] CH 8, CH 11 [DSA] CH 7, CH 10
Week 11 Nov 10	Problem 5 - Tower of Hanoi (OADD pg 455) HW 5 (available on 11/10) Search, Sort Recursion	Textbook: [OADD] CH 7, CH 10 [DSA] CH 11
Week 12 Nov 17	Quiz 5, HW 5 due (11/19) Graphs	Textbook: [OADD] CH 12, [DSA] CH 13
Week 13 Dec 1	Problem 6 - Path finding through a maze (OADD pg 755) HW 6 (available on 12/01) Graphs	Textbook: [OADD] CH 12, [DSA] CH 13 10 Tips for Clean Code
Week 14 Dec 8	Quiz 6, HW 6 due (12/10) Review	Textbook: 5-minute introduction to Git Michael Fudge
Week 15 Dec 15	**Final Exam**	Closed book, comprehensive, 2 hours

Part V: Assignment Weighting (How Your Final Grade is Being Calculated?)

Assessment Item	Percentage of final grade
Mid-term Exam	15%
Quiz & Other class assignments	15%
Final Exam (Comprehensive, closed book)	40%
Homework assignments	30%

Part VI: Delivery Mechanism

The following delivery mechanisms will be utilized:

Face-to-face lectures
Canvas: https://njit.instructure.com/courses/13334/
Online resources (other than iTunes)

1. Lectures -
   Lecture notes and supplemental references will be posted in Canvas weekly. In general, the lectures need not exactly follow the textbook, and should not be considered a replacement for the textbook.
   1. Introduce a new problem every week
   2. Lecture – 1 approaches the problem solving, associated data structure and algorithms
   3. Lecture – 2 will review any programming related aspects, if and as needed

2. Homework Assignments (30%)
   Programming exercises that ask you to develop programs based on material covered in class will be assigned on a regular basis, with writeups being posted to the course website. Some exercises will be completed in class with TA/instructor assistance, while others are to be completed at home. You will be assigned a grader at the beginning of the semester, and it is your responsibility to ensure that your work is submitted and graded by the stated due date for each lab.

3. Quizzes - (15%)
   There will be several graded quizzes during the semester. Each quiz is meant to review and test your knowledge of the material covered over several weeks. There is a time limit for each quiz administration. The questions for each quiz are typical of those that will be found on the final exam.

4. Participation & Mid-term exam - (15%)
   Participating during class, contributing answers / questions on Piazza, and engaging in class-related discussion with TAs / the instructor will earn you participation credit. Come mid-semester you'll be told where you stand on your participation score (and how to make up for a deficiency, if needed).

5. Final Exam - (40%)
   There will be a mid-term and a final exam. Questions for these exams will largely be drawn from and/or similar to the quizzes and other discussion questions.

   Unexcused late assignment submissions may not be accepted or accepted with penalty.

Part VII: Plagiarism and Academic Integrity

The approved "University Code on Academic Integrity" is currently in effect for all courses. Should a student fail a course due to a violation of academic integrity, they will be assigned the grade of "XF" rather than the "F" and this designation will remain permanently on their transcript.
All students are encouraged to look over the University Code on Academic Integrity and understand this document. Students are expected to uphold the integrity of this institution by reporting any violation of academic integrity. The identity of the student filing the report will be kept anonymous.
NJIT will continue to educate top tier students that are academically sound and are self-disciplined to uphold expected standards of professional integrity. **Academic dishonesty will not be tolerated at this institution.**

Part VIII: Getting Help - General

The IST Helpdesk is the central hub for all information related to computing technologies at NJIT. This includes being the first point of contact for those with computing questions or problems.
There are three ways to contact the Helpdesk:
1. Call 973-596-2900. Monday - Friday 8 am - 7 pm.
2. Go to Student Mall Room 48. Monday - Friday 8 am - 7 pm
3. Log a Help Desk Service Request online - https://ist.njit.edu/support/contactus.php.

Part IX: Getting Help – Canvas, WebEx etc.

PROTOCOL FOR DISTANCE LEARNING

- All instruction shall be delivered through Canvas. All directions and expectations for students will be posted on individual course Canvas pages. Students are responsible for checking each class on Canvas each day and for being especially attentive to email during this time.
- Attendance, participation, engagement, and understanding will all be monitored through submitted work.
- Work will be self-paced and guided by deadlines as designated. Coursework may be assigned and due in "chunks" so as to allow students to work through the material at their own pace while at home.
- All coursework will be submitted to Canvas.

Online participant etiquette

The following participant expectations should be the norm. Students who don't follow these guidelines can be removed from the Webex/Zoom meeting if necessary.

- Be on time
- Mute your microphone if you aren't talking
- Keep your video on throughout the class. In case of bandwidth issue, the instructor will turn off his video first.
- Raise virtual hand, if you want to speak on a point. You may also be asked specifically to comment on a topic.
- There will be online questions and/or surveys during the class that will be available for a limited duration. Please respond to them as and when asked. Some of these questionnaires will count towards the class participation grades.
- Only post chat messages relevant to the lessons