CSE 114: Introduction to Object-Oriented Programming
Course Information

Fall 2026 Lecture Section 1

Instructor: Dr. Paul Fodor

Stony Brook University

http://www.cs.stonybrook.edu/~cse114

Contents

- Course Description and Outcomes
- Instructor Information
- Class Time and Place
- Textbook
- Grading Schema
  - Re-grading policy
- Examinations
- Laboratory assignments
- Programming homework assignments
- Tentative Class Schedule
- Communications: Piazza and Email
- Student Accessibility Support
- Academic Integrity
- Critical Incident Management
- What do you need to get started in CSE114?

(c) Paul Fodor (CS Stony Brook)

Course Description

- "*An introduction to procedural and object-oriented programming methodology. Topics include program structure, conditional and iterative programming, procedures, arrays and records, object classes, encapsulation, information hiding, inheritance, polymorphism, file I/O, and exceptions.*

  Includes required laboratory."

  (https://www.cs.stonybrook.edu/students/Undergraduate-Studies/courses/CSE114)

  - Prerequisites: Level 4 or higher on the math placement exam.
  - **Advisory Prerequisite: CSE 101 or ISE108**
  - SBC: TECH
  - 4 credits (Lecture + Lab)

(c) Paul Fodor (CS Stony Brook)

Course Outcomes

- The following are the official course goals agreed upon by the faculty for this course:
  - An ability to program in an object oriented language, using concepts such as object classes, encapsulation, inheritance, and polymorphism.
  - An ability to use fundamental data structures such as arrays.
  - An ability to program with sound code structure and use systematic software debugging and testing techniques.
- These are adopted for our ABET accreditation.
  - https://www.cs.stonybrook.edu/students/Undergraduate-Studies/courses/CSE114

(c) Paul Fodor (CS Stony Brook)

Major Course topics

1. **Procedural Programming Basics:**
   - data types and variable declarations
   - assignment statements & expressions
   - input/output
   - textual manipulation & strings
   - conditional (branching) statements
   - iteration = loops and recursion
   - method construction and invocation

(c) Paul Fodor (CS Stony Brook)

Major Course topics

2. **Arrays:**
   - **collect data in arrays**
   - **indexed variables**
   - **array manipulations**
   - **searching**
   - **sorting**

(c) Paul Fodor (CS Stony Brook)

Major Course topics

3. **Object Oriented Programming:**
   - designing and constructing classes
   - instantiating objects
   - inheritance and polymorphism
   - abstract classes
   - interfaces
4. **Recursion**
   - using recursion to solve problems

(c) Paul Fodor (CS Stony Brook)

Instructor Information

- Dr. Paul Fodor
  214 New Computer Science Building
  - Office hours: Tuesdays and Wednesdays 11AM-12:30PM in the New Computer Science Building, room 214.
- Email: Paul (dot) Fodor (at) stonybrook (dot) edu
  - Please include "CSE 114" in the email subject and your name in your email correspondence
- Phone: 1 (631) 632-9820

(c) Paul Fodor (CS Stony Brook)

Class Time and Place

- **CSE 114-01 Lecture section 1:** TuTh 12:30PM - 1:50PM, Light Engineering 102.
  - **CSE 114-L01 Lab section 1:** MoWe 9:30AM - 10:50AM, Old Computer Science 2114.
  - **CSE 114-L02 Lab section 2:** MoWe 3:00PM - 4:20PM, Old Computer Science 2129.
  - **CSE 114-L03 Lab section 3:** MoWe 11:00AM - 12:20PM, Computer Science 2129.
  - **CSE 114-L04 Lab section 4:** MoWe 8:00AM - 9:20AM, Computer Science 2311.
    - Note: The students are allowed to attend any lab section for our CSE114 lecture section 1 (for the same lab number). That is, if you miss your lab section, then you can go with a later lab section for the same lab number.

(c) Paul Fodor (CS Stony Brook)

Textbook

- **Optional:** Introduction To Java Programming and Data Structures, Author: Y. Liang, Publisher: Pearson, 13th edition (August 2023) or 12th edition (December 4, 2019). Older versions are also fine since the textbook does not change much.

  https://www.amazon.com/Introduction-Programming-Structures-Comprehensive-Version/dp/0136520235/

(c) Paul Fodor (CS Stony Brook)

Grading Schema

- Grades will be based on homework and exams according to the following formula:
  - Homework assignments -- 15%
  - Labs -- 10%
  - Midterm exams (2) -- 50% (25% each)
  - Final exam -- 25%

(c) Paul Fodor (CS Stony Brook)

Grading Schema

- Default Grade Cutoffs:
  - A [95-100], A- [90-95),
    B+ [87-90), B [83-87), B- [80-83),
    C+ [77-80), C [73-77), C- [70-73),
    D+ [65-70), D [60-65), F [0-60)
    - The Pass/No Credit (P/NC) option is not available for this course: This policy applies to all CSE/ISE undergraduate courses used to satisfy the graduation requirements for the major.

(c) Paul Fodor (CS Stony Brook)

Grading Schema

- SPECIAL RULE: If your final total in the class is above the class average, you're guaranteed to receive a grade of C or higher for this class.

(c) Paul Fodor (CS Stony Brook)

Grading Schema

- SPECIAL RULE **details**: at the end of the semester after all points from the exams, homework assignments and labs are incorporated in the final total for all students, I will compare the class average with the default cutoff for a C grade (that is, 73).
  - If the class average is above 73, then I use the default grade cutoffs, that is:

A [95-100], A- [90-95), B+ [87-90), B [83-87), B- [80-83), C+ [77-80), C [73-77), C- [70-73), D+ [65-70), D [60-65), F [0-60).

  - If the class average is below 73 (for example 70.91), then I deduct the difference between the fixed cutoff and the integer part of the class average from all grades cutoffs.
    - For example, if the class average is 70.91, then the grade cutoffs will become: A [92-100], A- [87-91), B+ [84-87), B [80-84), B- [77-80), C+ [74-77), C [70-74), C- [67-70), D+ [62-67), D [57-62), F [0-57).

(c) Paul Fodor (CS Stony Brook)

Grading Schema

- SPECIAL RULE note: I taught this class every year since 2011. Although assignments and exams were similar in difficulty, the total average fluctuates from semester to semester.
  - The average of all semesters between 2011 and now was 73.57

(c) Paul Fodor (CS Stony Brook)

Grading Schema

- **Notes on grading at the end of the semester:** After we grade the final exam, we will post the final exam grades on Brightspace and final exam statistics on Piazza. We announce in-person office hours to review the final exams on Piazza and hold them within the next 24 hours. After the final exam reviewing office hours, we will compute and post on Piazza the average and the final grades' cutoffs, then compute the final grades for all students and post them on Brightspace. We will post the final grades on SOLAR immediately.

(c) Paul Fodor (CS Stony Brook)

Grading Schema

- **Re-grading:** For re-grading of an assignment or exam, arrange a re-evaluation **within one week** of receiving the graded work.
  - All requests that are later than one week from the date the graded work is posted on Brightspace (and exams are returned in class) will not be entertained.
  - All regrading is done either in the instructor office hours or by email to the instructor.

- All grades will be posted on Brightspace: http://Brightspace.stonybrook.edu for privacy reasons.

(c) Paul Fodor (CS Stony Brook)

Grading

- Finally, the final grade you receive in this class will reflect, as far as possible, the extent to which you have mastered the concepts and their applications.
  - How much someone needs a grade, or how close they are to the next higher grade, will have no effect on grade.
  - As the instructor, I want everyone to do well in this course, and will make every reasonable effort to help you understand the material taught.
  - However, the grades provided at the end of the semester are final, except for rare situations involving grading errors.
  - They will not be altered for any reason, so please do not ask me to do so.

(c) Paul Fodor (CS Stony Brook)

Examinations

- There will be two midterm exams and a final exam. All exams will be closed-book and closed-notes. Students must bring photo ID to each exam. Students will not be admitted late to any exam.
  - Midterm Exam #1:Thu. 10/1/2026, closed book in-person exam, in classroom, during class time (80 minutes).
  - Midterm Exam #2:Thu. 11/5/2026, closed book in-person exam, in classroom, during class time (80 minutes).
  - Final Exam: Tuesday, December 15, 11:15am-1:15pm (120 minutes), closed book in-person exam, in classroom (according to the university final exams' schedule for our lecture time slot).
- Do not miss any exams. Make-up exams will be given only in extenuating circumstances (e.g., in-person doctor's note stating that you were ill and unfit to take the exam).

(c) Paul Fodor (CS Stony Brook)

Lab exercises

- You will be given problems that require a programmed solution during the laboratory.
  - 0 - the student did not attend the lab,
  - 3 - the solutions are complete OR the student spent the entire lab solving the required lab problems (in this case, the students **may not arrive at the lab after the lab has started** and may not leave until the lab ends or they show complete solutions for all the lab problems).
- The students must use their own laptops (recommended).
  - The university runs a Student Laptop Loaner Program: https://it.stonybrook.edu/services/student-laptop-loaner-program

(c) Paul Fodor (CS Stony Brook)

Lab exercises

- **Missing some labs:**
  - If you have a valid reason (e.g., doctor's note with an ailment for the day of the lab, athletic event recognized by the school), then I can accept the lab by email to the instructor (we cannot accept labs (late or not) by email in any other case because the students are required to attend the labs).
  - We waive 2 labs from everyone in the class for the entire semester for other reasons, so, if you have a personal event and you cannot attend the lab, then you can miss/skip up to 2 labs without a grade penalty (this is automatically done at the end of the semester, so you do not need to let the instructor know since we automatically waive 2 labs for everyone in the class).

(c) Paul Fodor (CS Stony Brook)

Lab exercises

- **How the Teaching Assistants (TAs) run the lab:**
  - I post the lab assignments (2-5 problems) on Brightspace and Piazza before the lab.
  - The lab TAs show the lab description on Piazza, read each problem description loudly, then implement it from scratch (while describing why it is coded this way and how it works), so that the students can follow and learn how to solve problems themselves (questions are welcome).
    - Once a program is implemented, they show it on the projector and help all the students implement it. Once all the students finish the implementation, the TAs continue to the next problem.

(c) Paul Fodor (CS Stony Brook)

Lab exercises

- Due to how the labs are scheduled (some are before and some are after the lectures) and that at the beginning of the semester we need to cover material so that the labs are meaningful, the lab material might sometimes be slightly ahead of the lecture. I will try to minimize such cases!
  - In such cases following the lab might be difficult, but students copy the program and learn by **reviewing** the code, following the demonstration and **practicing programming**. It is still very beneficial for students.
    - In fact, there is a software development practice called ***pair programming***, where two developers collaborate at one workstation, one typing and the other reviewing.

(c) Paul Fodor (CS Stony Brook)

Lab exercises

- Sometimes the TAs cannot cover all the problems that I assign in the lab. That is also fine! Programming requires a lot of practice, so I view these remaining problems as valuable opportunities for independent discovery.
  - You do not need to finish the problems in the lab. **You still get the complete credit for the lab if you stay until the end of the lab session!** The students aren't "behind" if they don't finish in the session. You can still ask questions about these problems later (e.g., in office hours or on the Piazza discussion board).
  - Wrestling with a challenge outside of guided hours is often where the most significant 'aha!' moments happen, helping you build the intuition and resilience needed for real-world development. I intentionally provide extra material to ensure you never run out of ways to sharpen your skills (I also want students to come to the lab instead of just showing the solutions and leaving). These problems aren't meant to overwhelm you, but rather to give you a sandbox where you can apply what you've learned at your own pace. Mastery comes from repetition and trial-and-error.

(c) Paul Fodor (CS Stony Brook)

Assignments

- Homework assignments due on fixed dates and times
  - **no late submission is permitted**
- All assignments should be submitted electronically
  - Brightspace
    - We cannot accept submissions by email at any date or time
    - All assignments are graded after the assignment deadline and the final grade will be posted on Brightspace.

(c) Paul Fodor (CS Stony Brook)

Tentative Class Schedule

Week	Lecture Topics
1	Introduction to Computers
2	Programming and Java, Elementary Programming
3	Selections
4	Mathematical Functions, Characters, and Strings
5	Loops
6	Methods
7	Arrays
8	Multi-dimensional Arrays
9	Objects and Classes
10	Object-Oriented Thinking
11	Inheritance and Polymorphism
12	Exception Handling and Text I/O
13	Abstract Classes and Interfaces
14	Recursion

(c) Paul Fodor (CS Stony Brook)

Class Communication: Piazza

- The Piazza discussion board should be used for all communication with the teaching staff for questions about the course assignments and material
  - Piazza is also used for class announcements.
  - Piazza is a forum for additional learning and assistance
  - You are expected to use the Piazza forum for all non-personal, course-related communication
    - Like questions about what a homework problem is asking, technical problems that need troubleshooting, or other questions that might be of interest to other students must be posted to Piazza and not emailed to the instructor or a TA

(c) Paul Fodor (CS Stony Brook)

Piazza

- The following are NOT appropriate uses of Piazza:
  - cyber-bullying
  - posting memes
  - posting more than a few lines of source code from an attempt at a homework problem
  - posting the solution to a homework problem or a link to a website containing the solution
  - in general, anything unrelated to the course material and student learning
- Anonymous posting is turned off, so we can see who you are.
  - If you use Piazza inappropriately, you will be removed from Piazza.

(c) Paul Fodor (CS Stony Brook)

Email Etiquette

- If you have a private matter to discuss, when emailing your instructor about the course, use the following guidelines to ensure a timely response:
  - use your official **@stonybrook.edu** email account (we cannot respond to other emails due to FERPA regulations)
  - use a descriptive subject line that includes "CSE114" and a brief note on the topic
  - begin with a proper greeting, such as "Hi Prof. Fodor"
  - briefly explain your question or concern or request including the course (I am teaching several courses)
  - end with a proper closing that includes your full name and SBU ID number

(c) Paul Fodor (CS Stony Brook)

Student Accessibility Support Center Statement:

https://www.stonybrook.edu/commcms/provost/faculty/handbook/academic_policies/syllabus_statement.php

- If you have a physical, psychological, medical, or learning disability that may impact your course work, please contact the Student Accessibility Support Center, Stony Brook Union Suite 107, (631) 632-6748, or at sasc@stonybrook.edu. They will determine with you what accommodations are necessary and appropriate. All information and documentation is confidential.
- Students who require assistance during emergency evacuation are encouraged to discuss their needs with their professors and the Student Accessibility Support Center. For procedures and information go to the following website: https://ehs.stonybrook.edu//programs/fire-safety/emergency-evacuation/evacuation-guide-disabilities and search Fire Safety and Evacuation and Disabilities.
- **All documentation of disability is confidential.**

(c) Paul Fodor (CS Stony Brook)

Academic Integrity Statement:

- The following rules are posted in every course syllabus: "*Each student must pursue his or her academic goals honestly and be personally accountable for all submitted work. Representing another person's work as your own is always wrong.* ***Any suspected instance of academic dishonesty will be reported to the Academic Judiciary.*** *For more comprehensive information on academic integrity, including categories of academic dishonesty, please refer to the academic judiciary website at* http://www.stonybrook.edu/commcms/academic_integrity/"

(c) Paul Fodor (CS Stony Brook)

Academic Integrity

- You can discuss general assignment concepts with other students: explaining how to use systems or tools and helping others with high-level design issues
- You **MAY NOT share** assignments, source code or other answers by copying, retyping, looking at, or supplying a file
  - Assignments are subject to manual and automated similarity checking (We do check! and our tools for doing this are much better than cheaters think)
- If you cheat, you will be brought up on academic dishonesty charges - we follow the university policy:
  - http://www.stonybrook.edu/uaa/academicjudiciary

(c) Paul Fodor (CS Stony Brook)

Academic Integrity

- Anything that you hand in, whether it is a written problem or a computer program, must be written by you in your own words. If you base your code on any other solution written by someone else or you use an AI to generate any part of your code, you are cheating.

(c) Paul Fodor (CS Stony Brook)

Examples of Academic Dishonesty

- Unpermitted collaboration (on a paper, homework, lab reports, etc.). Unless an instructor has explicitly approved working together, students should assume, for their own protection, that it is not permitted.
- Helping someone else to plagiarize from one's own homework (for example, by giving them a copy of yours, or doing it for them)
  - This includes having a public repository on Github that other students can copy from.
- Representing someone else's source code as one's own. If another person's code is being used, it must be properly cited.
- Buying or selling source code.
- Using source code from the internet without properly citing the source.

(c) Paul Fodor (CS Stony Brook)

Academic Dishonesty

- The instructor makes a recommendation at the Academic Judiciary office
  - Cheating is cheating! No matter the amount of cheating or if one is the source or destination of cheating.
  - Do not cheat! You are cheating yourself.
  - Our job is the teach you the material and make sure that you learn it.
  - Our recommendation is always F for the cheaters!

(c) Paul Fodor (CS Stony Brook)

Critical Incident Management:

- Stony Brook University expects students to respect the rights, privileges, and property of other people. Faculty are required to report to the Office of Student Conduct and Community Standards any disruptive behavior that interrupts their ability to teach, compromises the safety of the learning environment, or inhibits students' ability to learn. Faculty in the HSC Schools and the School of Medicine are required to follow their school-specific procedures. Further information about most academic matters can be found in the Undergraduate Bulletin, the Undergraduate Class Schedule, and the Faculty-Employee Handbook.

(c) Paul Fodor (CS Stony Brook)

Catastrophic events

- Major illness, death in family
- Formulate a plan (with your CEAS academic advisor) to get back on track as soon as possible
- Advice
  - Once you start running late, it's really hard to catch up

(c) Paul Fodor (CS Stony Brook)

What do you need to get started?

- Brightspace account
  - http://Brightspace.stonybrook.edu
- Course Web page: http://www.cs.stonybrook.edu/~cse114
  - Lecture notes and schedule
  - Syllabus
- Eclipse IDE: https://www.eclipse.org
  - You should download the Eclipse IDE for Java Developers (cost: free)
    - Latest Eclipse comes with JDK, so you don't need to install it separately
  - Learn to use the debugger in Lab 1!!!

(c) Paul Fodor (CS Stony Brook)

Java: How does it work?

- Java Source Code:
  - you write code in `ClassName`.***java*** **text files**

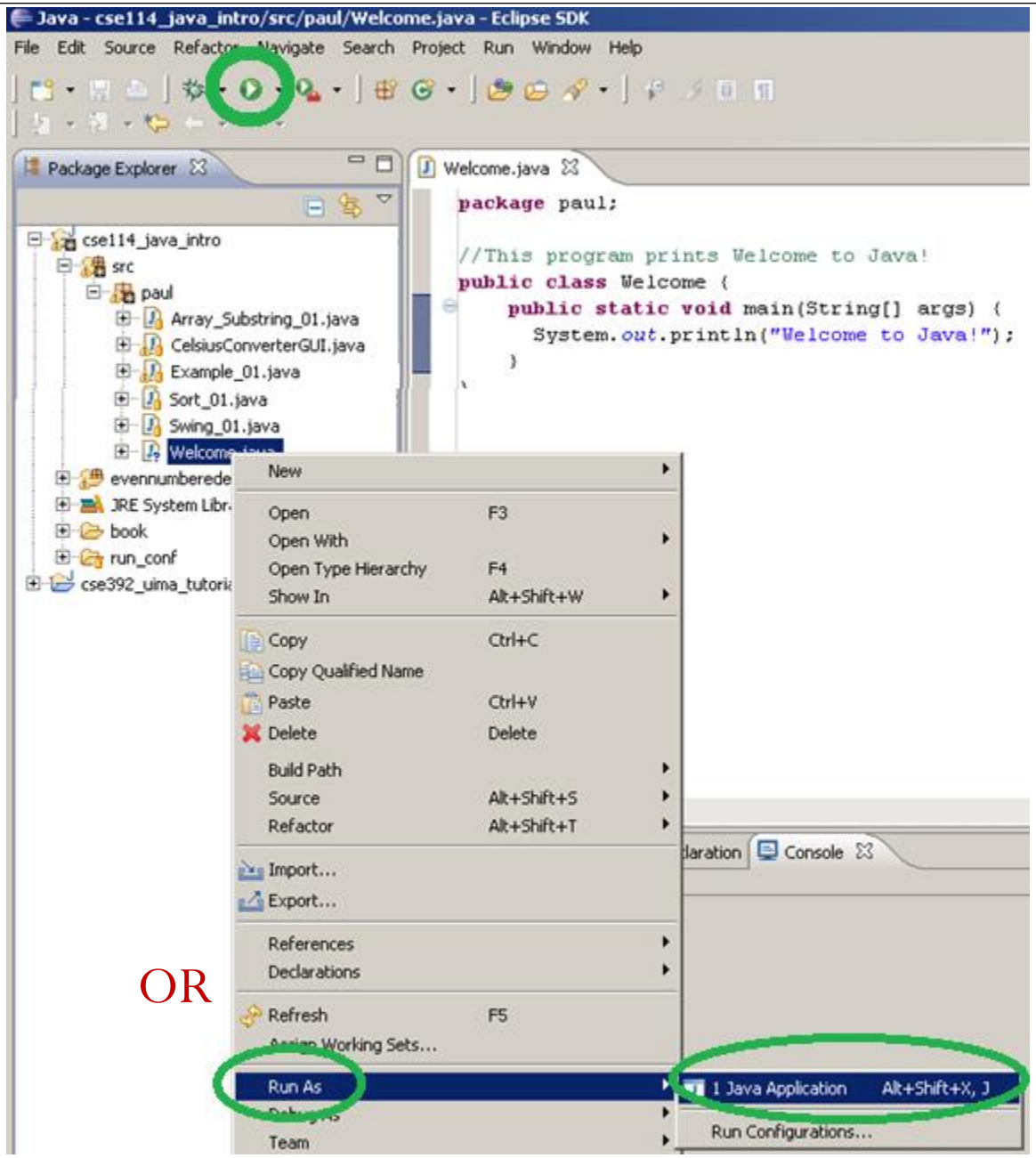
(c) Paul Fodor (CS Stony Brook)

Tools for Writing Java Programs

- 1st Approach – the bare minimum
  - edit Java source code in text editor (ex: Notepad, Pico,…)
  - compile source code into class files from command line: javac
  - run the Java bytecode with the Java Virtual Machine: java
  - can be tedious and poor interactivity
- 2nd Approach – Integrated Development Environment (IDE)
  - combines writing, compiling, running and debugging Java code into a single application
  - makes coding much more efficient and organized
  - Eclipse, NetBeans, IntelliJ IDEA, VisualCode, etc.
    - We can help with Eclipse, but you can use other IDEs on your own

(c) Paul Fodor (CS Stony Brook)

Running Programs in Eclipse



(c) Paul Fodor (CS Stony Brook)

Command line execution

- Java Source Code:
  - you write code in `ClassName`***.java*** **files**
- Compile your Program:

  `javac` *ClassName.java*

  The Result is: Java Executable Code (bytecode)
  - *ClassName.class* files = Java bytecode (binary, not humanly readable)
- Now you can run your java program using the Java Virtual Machine (JVM):

  `java` *ClassName*

(c) Paul Fodor (CS Stony Brook)

Please

- Please be on time
- Please show respect for your classmates
- Please turn off (or use vibrate for) your cellphones

...

- On-topic questions are welcome

(c) Paul Fodor (CS Stony Brook)

Welcome and Enjoy!