CSE 260, Computer Science B
Course Information

Fall 2017
Stony Brook University
Instructor: Dr. Paul Fodor

http://www.cs.stonybrook.edu/~cse260
Course Description

• “Second part of a two-semester sequence, CSE 160 and CSE 260. Further development of the object-oriented design strategies presented in CSE 160. Continues with introductions to event-driven programming, graphical user interfaces, and design patterns. Includes an extended design and programming project. (https://www.cs.stonybrook.edu/students/Undergraduate-Studies/courses/CSE260)

• Prerequisites: CSE 160

• Corequisite: CSE 261
Course Outcomes

- The following are the official course goals agreed upon by the faculty for this course:
  - An ability to apply techniques of object-oriented programming in the context of large-scale programs.
  - An understanding of the importance of time and memory efficiency in algorithm and data structure design.
  - An ability to use recursion to solve programming problems.
  - An ability to use programming tools such as syntax-directed editors, debuggers, execution profilers, documentation generators, and revision-control systems.
Major Topics Covered in Course

- Software development process, specifications and requirements, object-oriented design, version management.
- Event-driven programming, applets.
- Graphical user interfaces.
- Design Patterns, Threads and synchronization.
- Abstract data types, collection types, mapping types.
- Linked lists, iterators.
- Stacks and applications.
- Queues and applications.
- Recursion.
- Trees, binary search trees, heaps and balanced trees.
- Searching, binary search, hash tables, complexity, big-O notation, using profiling tools to assess program performance.
- Sorting, mergesort, heapsort, quicksort.
Instructor Information

• Dr. Paul Fodor
  214 New Computer Science Building
• Office hours: Mondays and Wednesdays 5:30PM-7:00PM
• Phone: 1 (631) 632-9820
• Email: paul (dot) fodor (at) stonybrook (dot) edu
  • Please include “CSE 260” in the email subject and your name in your email correspondence
• TAs: see course Web page:  
  http://www.cs.stonybrook.edu/~cse260
• **Blackboard** will be used for assignments, grades and course material.
Class Time and Place

- CSE 260-01 (91193) Computer Science B: Honors (Lecture): MoWe 4:00PM - 5:20PM, Frey Hall 309.

- CSE 260-L01 (91926) Computer Science B: Honors (Laboratory): TuTh 4:00PM - 5:20PM, Frey Hall 305.
Textbooks


Grading Schema

• Grades will be based on homework and exams according to the following formula:
  • Homework assignments = 12.5%
  • Project phase 1 = 2.5%
  • Project phase 2 = 2.5%
  • Project phase 3 (final) = 7.5%
  • Quizzes = 5%
  • Labs = 10%
  • Midterm exam 1 = 20%
  • Midterm exam 2 = 20%
  • Final exam = 20%
Examinations

- Midterm 1: We. 10/4, 4:00PM - 5:20PM (80 minutes).
- Midterm 2: We. 11/8, 4:00PM - 5:20PM (80 minutes).
- Final Exam: Tuesday, December 12, 2017, 8:30-10:30 PM (120 minutes).
  - See Final Exams University Schedule for final exam schedules: http://www.stonybrook.edu/registrar-finals.shtml
  - The exams will be like the problems that we solve in the class!
- 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.
Grading

- **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]

- **SPECIAL RULE:** If all your grades, including homework assignments, quizzes, recitation and your three exam grades are above the respective class averages, you're guaranteed to receive a grade of C or higher for this class.

- There will be extra credit problems as a part of quizzes and homework assignments which values to an increase of less than 4% in the final grade.

- There will be in-class quizzes / brief assessments used to practice the class material and measure growth in knowledge, abilities, and skills. They will be solved in class and they are valued 2 points each.
Grading

• 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.
Assignments

• Homework assignments due on fixed dates and times.
  • no late submission is permitted
• All assignments should be submitted electronically on Blackboard
Lab exercises

- You will be given problems that require a programmed solution during lab hours. You will have only the lab period to edit, compile, and execute your solution.

- Attendance is mandatory, if you want credit
  - you can leave early only if you showed all your assigned work
  - if you come later than 15 minutes from the start of the lab, then you don't get the credit for the lab

- Demonstrate your work to Lab-TA before you leave for lab credit
  0 – 3 points:
  - 0 - Student did not attend the lab or program does not even compile.
  - 1 - Student attended the lab, program compiles but has major problems.
  - 2 - Student attended the lab, and program partially works (with some minor errors)
  - 3 - Student attended the lab, and program is correct
Regrading of Homework/Exams

• Please meet with a grading TA or the instructor and arrange for regrading.

• You have one week from the day grades are posted or mailed or announced!

• Late requests will not be entertained
<table>
<thead>
<tr>
<th>Week</th>
<th>Lecture Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Generics in Java</td>
</tr>
<tr>
<td>2</td>
<td>Lists, Stacks, Queues, and Priority Queues in Java</td>
</tr>
<tr>
<td>3</td>
<td>Sets and Maps in Java</td>
</tr>
<tr>
<td>4</td>
<td>Developing Efficient Algorithms</td>
</tr>
<tr>
<td>5</td>
<td>Sorting</td>
</tr>
<tr>
<td>6</td>
<td>Implementing Lists, Stacks, Queues, and Priority Queues</td>
</tr>
<tr>
<td>7</td>
<td>Binary Search Trees</td>
</tr>
<tr>
<td>8</td>
<td>AVL Trees</td>
</tr>
<tr>
<td>9</td>
<td>Hashing</td>
</tr>
<tr>
<td>10</td>
<td>Graphs and Applications</td>
</tr>
<tr>
<td>11</td>
<td>Weighted Graphs and Applications</td>
</tr>
<tr>
<td>12</td>
<td>2-4 Trees and B-Trees</td>
</tr>
<tr>
<td>13</td>
<td>Red-Black Trees</td>
</tr>
<tr>
<td>14</td>
<td>Multithreading and Parallel Programming</td>
</tr>
<tr>
<td>15</td>
<td>Design patterns</td>
</tr>
</tbody>
</table>
Disability Support Services

• If you have a physical, psychological, medical or learning disability, contact the DSS office at Room 128 ECC. Phone 632-6748/TDD

• If you are planning to take an exam at DSS office, you need to tell me ahead of time for every exam.

• All documentation of disability is confidential.
Academic Integrity

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/](http://www.stonybrook.edu/commcms/academic_integrity/)"
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](http://www.stonybrook.edu/uaa/academicjudiciary)
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 or pieces of a paper from the internet without properly citing the source**.
Academic Integrity

- 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!
Catastrophic events

- Major illness, death in family
- Formulate a plan (with your CEAS academic advisor) to get back on track

Advice
- Once you start running late, it’s really hard to catch up
What do you need to get started?

- Blackboard account
  - [http://blackboard.stonybrook.edu](http://blackboard.stonybrook.edu)
- Java JDK standard edition:
  - [http://www.oracle.com/technetwork/java/javase/downloads](http://www.oracle.com/technetwork/java/javase/downloads)
- Eclipse IDE:
- Learn to use the debugger!!!
- Liang’s student Web site:
  - [http://www.cs.armstrong.edu/liang/intro11e](http://www.cs.armstrong.edu/liang/intro11e)
Software

• Necessary Software:
    • You should download JDK for your operating system (cost: free)
  • Eclipse IDE: [http://www.eclipse.org](http://www.eclipse.org)
    • You should download the Eclipse IDE for Java Developers (cost: free)
    • 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, etc.
Individual projects

• "An extended design and programming project."

• [https://www.cs.stonybrook.edu/students/Undergraduate-Studies/courses/CSE260](https://www.cs.stonybrook.edu/students/Undergraduate-Studies/courses/CSE260)

  • Project phase 1: Monday, October 9 (proposal in class)
  • Project phase 2: Monday, November 13 (presentations in class)
  • Project phase 3 (final): Monday, December 4 (presentations in class)
Sample project: Go
Another sample project: Artificial Intelligence (AI) Poker
Another sample project: AI Blackjack
Pass Line

Don't Pass Line

$1 $2 $3 $4 $5

Point 10

Good Luck

Roll Dices
Another sample project: Baccarat
Another sample project: Pai Gow Poker (double-hand poker)
Another sample project: 24 Game
Another sample project: Bridge
Another sample project: Go Fish

(c) Paul Fodor (CS Stony Brook)
Another sample project: Wheel of Fortune
Another sample project: Cosmic Wimpout

http://cosmicwimpout.com/p/7/How-to-play
Another sample project: Scrimish
Another sample project: UNO!
Another sample project: the rush-hour game

http://www.thinkfun.com/play-online/rush-hour/
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
Welcome and Enjoy!