CSE 160, Computer Science A: Honors
Course Information

Spring 2019
Stony Brook University
Instructor: Dr. Paul Fodor

http://www.cs.stonybrook.edu/~cse160
Course Description

“First part of a two-semester sequence, CSE 160 and CSE 260. Emphasizes a higher-level, object-oriented approach to the construction of software. Focus on software engineering issues such as programming style, modularity, and code reusability. Includes the way in which software tools can be used to aid the program development process. First considers the construction of small programs, continues by treating the design and implementation of program modules, and culminates in an introduction to object-oriented design techniques suitable for larger programs.”

(https://www.cs.stonybrook.edu/students/Undergraduate-Studies/courses/CSE160)

Prerequisites: Computer Science Honors Program or Honors College or WISE program or University Scholar or permission of instructor
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 understanding the importance of programming style and modularity to the construction and evolution of robust software.
- An ability to use programming tools such as syntax-directed editors, debuggers, documentation generators and testing frameworks.
- An ability to construct software modules consisting of several hundred lines of code.
Course Focus

- Introduction to programming (in Java):
  - conditional statements
  - loops
  - methods
- Fundamental data structures of high-level programming: arrays, lists, stacks
- Algorithms
- Basic concepts of object-oriented programming
  - object classes
  - encapsulation
  - inheritance
  - polymorphism
  - Application: GUIs
- Programming assignments
  - systematic software debugging
  - systematic testing techniques
Major Course topics

- Objects, methods, and classes
- Object state, mutability and immutability, object interaction
- Accessors and mutators
- Fields, parameters, and local variables
- Abstraction and modularization
- Arrays, collections, and iterators
- Library classes, documentation, using the Javadoc documentation generator
- Designing classes: modularity and information hiding, cohesion, coupling, refactoring
- Programming style: naming, commenting, formatting, avoiding code repetition, using a syntax-directed editor
Major Course topics

- Testing and debugging: using a debugger, unit testing, regression testing, using the JUnit testing framework
- Regular expressions and programming applications
- Recursion
- Inheritance, polymorphism, overloading and overriding, static and dynamic type, subtyping, dynamic method lookup
- Abstract classes and interfaces
- Handling errors: exceptions and handlers
- Files and input/output, streams, readers and writers, character sets
- Introduction to graphical user interfaces, using the Swing framework
- Introduction to object-oriented design, using CRC cards
Instructor Information

- Dr. Paul Fodor
  214 New Computer Science Building
- Office hours: Mondays and Tuesdays 3:30-5pm
- Email: paul (dot) fodor (at) stonybrook (dot) edu
  - Please include “CSE 160” in the email subject and your name in your email correspondence
- TAs: see course Piazza
General Information

• Course Web page: http://www.cs.stonybrook.edu/~cse160

• **Blackboard** will be used for assignments, grades and course material.
General Information

• Meeting Information (Class Time and Place):
  • CSE 160 Computer Science A: Honors (Lecture): TuTh 5:30-6:50PM, CS 2129.
  • CSE 161 Laboratory for Computer Science A: Honors: TuTh 7-8:20PM, CS 2129.


• MyProgrammingLab.com is the online testing system that comes with the textbook and we will use it for online computer programming exercises executed in the laboratory and at home as part of the homework assignments.

• The book with ISBN 9780133813470 includes the subscription to MyProgrammingLab.

• Students who wish to purchase access to MyProgrammingLab without the textbook may do so by visiting the MyProgrammingLab.com website provided by the Pearson publisher of our textbook.


• This is the comprehensive version of the brief version above.

• We will use material from this textbook in CSE260, so it could be the textbook of both courses.
Software

• Necessary Software:
    • You should download JDK for your operating system (cost: free)
  • Eclipse IDE: http://www.eclipse.org
    • You should download the Eclipse IDE for Java Developers (cost: free)
Coursework

• 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%
  • CSE161 is the lab for CSE160 and will get the same grade as CSE160.
Important Dates

• Midterm Exam #1: Tuesday, March 5, 2019, during class time (80 minutes).
• Midterm Exam #2: Evening exam, Tuesday, April 30, 2019, during class time (80 minutes).
• Final Exam: Wednesday May 15, 2019, 5:30-7:30pm (2 hours final exam), in classroom.
  • See Final Exams University Schedule for exam schedules: http://www.stonybrook.edu/registrar/finals.shtml
  • The exams will be like the problems that we solve in the class!
Assignments

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

• Simple Coding Exercises done in Computer Science (CS)
  • You have only the lab-hour 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 late, then you don't get the credit for the lab
      • you can come to another lab that same day to get the credit for the lab () same rules apply as above!
  • 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
# Class Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Lecture Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to Computers, Programming and Java</td>
</tr>
<tr>
<td>2</td>
<td>Elementary Programming and Selections</td>
</tr>
<tr>
<td>3</td>
<td>Mathematical Functions, Characters, and Strings, Loops</td>
</tr>
<tr>
<td>4</td>
<td>Methods</td>
</tr>
<tr>
<td>5</td>
<td>Arrays and Multi-dimensional Arrays</td>
</tr>
<tr>
<td>6</td>
<td>Midterm Review and Exam 1</td>
</tr>
<tr>
<td>7</td>
<td>Objects and Classes, Object-Oriented Thinking</td>
</tr>
<tr>
<td>8</td>
<td>Spring recess</td>
</tr>
<tr>
<td>9</td>
<td>Inheritance and Polymorphism</td>
</tr>
<tr>
<td>10</td>
<td>Exception Handling and Text I/O</td>
</tr>
<tr>
<td>11</td>
<td>Abstract Classes and Interfaces</td>
</tr>
<tr>
<td>12</td>
<td>JavaFX Basics, Event-Driven Programming</td>
</tr>
<tr>
<td>13</td>
<td>JavaFX UI Controls and Multimedia</td>
</tr>
<tr>
<td>14</td>
<td>Midterm exam 2, Recursion</td>
</tr>
<tr>
<td>15</td>
<td>Recursion</td>
</tr>
</tbody>
</table>
Student Accessibility Support Center

• If you have a physical, psychological, medical or learning disability that may impact your course work, please contact Student Accessibility Support Center, ECC (Educational Communications Center) Building, Room 128, (631)632-6748. They will determine with you what accommodations, if any, 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 Student Accessibility Support Center. For procedures and information go to the following website:
  http://www.stonybrook.edu/ehs/fire/disabilities

• 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/"
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
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 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
Piazza

- The following are NOT appropriate uses of Piazza:
  - cyber-bullying
  - posting memes
  - complaining about a grade
  - airing concerns/comments/criticisms about the course
  - 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.
Email Etiquette

- 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 an other email due to FERPA regulations)
  - use a descriptive subject line that includes "CSE160" 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 (we are teaching several courses)
  - end with a proper closing that includes your full name, Net ID and SBU ID number
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/intro10e](http://www.cs.armstrong.edu/liang/intro10e)
Past Projects: Artificial Intelligence (AI) Poker
AI Blackjack
Craps

Pass Line

Don't Pass Line

$1  $2  $3  $4  $5

Point 10

Good Luck

Roll Dices
Baccarat

User - purse=20; sum=0

Computer - sum=0

Round 1  Action: continue

Bet: $1  $2  $3  $4  $5  Bet: $0

Action: Continue

Result:  => New purse: $  Next
Pai Gow Poker (double-hand poker)
24 Game
Bridge
Go Fish
Wheel of Fortune
Cosmic Wimpout

http://cosmicwimpout.com/p/7/How-to-play
Scrimish
UNO!
Tools for Writing Java Programs

- 1\textsuperscript{st} Approach – the bare minimum
  - edit Java source code in text editor (ex: Notepad or Pico)
  - compile source code into class files from command line: javac
  - can be tedious
  - poor interactivity

- 2\textsuperscript{nd} 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, etc.
Java: How does it work?

- **Java Source Code**
  - you write `????????.java` files

- Compile your Program
  
  `javac` `????????.java`
  
  OR

- **Build menu option in the Run menu** included in the Eclipse IDE

- The Result is: **Java Executable Code (bytecode)**
  - `????????.class` files = Java bytecode - not humanly readable

- Now you can run your java program using the **Java Virtual Machine (JVM)**:
  
  `java` `YourProgramName`
  
  OR **Run** button included in the Eclipse IDE
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!