CSE 114, Computer Science 1
Course Information

Summer 2018
Online on SBConnect
Stony Brook University
Instructor: Dr. Paul Fodor

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

• “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. Software debugging and testing techniques are emphasized. Includes required laboratory.”

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

• Prerequisites: Level 4 or higher on the math placement exam (or MAT 123+)

• Advisory Prerequisite: CSE 101 or ISE108.
Course Description

- Online course
  - This online course provides the students with the flexibility and convenience of studying online for a course with very high demand in order to support timely degree completion and meet the student’s academic goal.
  - The course is taught online, synchronously and asynchronously delivered, and provides all materials, labs, discussions, advising and contact with faculty online over the Internet.
  - This course includes required online synchronous laboratory.
Course Description

- The students are required to go to Stony Brook University to take the course exams:
  - **Midterm Exam:** Thursday, June 14, 2018, 12:00PM-2:00PM, during class time, Frey 104.
  - **Final Exam:** Thursday, July 5, 2018, 12:00PM-2:00PM, during class time, Frey 104.
A successful online student should possess some specific qualities:

- Strong self-discipline,
- A high level of professional, personal, or educational motivation;
- Good time management skills;
- Ability to work and study independently without face to face interaction with faculty and classmates to complete the work;
- Experience using computers and the Internet;
- Access to a computer and the Internet to work many hours a week online;
- Have already completed all the prerequisites;
- Understand that an online course requires as many hours as an on-campus course — or more.
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.
Course Focus

- Introduction to programming (in Java):
  - conditional statements
  - loops
  - methods

- Fundamental data structures of high-level programming: arrays, lists, stacks, ...

- Basic concepts of object-oriented programming
  - object classes
  - encapsulation
  - inheritance
  - polymorphism
  - Application: GUIs

- Algorithms

- Programming assignments
  - systematic software debugging techniques
  - systematic testing techniques
1. **Procedural Programming Basics:**
   - variable declarations
   - data types
   - assignment statements & expressions
   - textual manipulation & strings
   - input/output
   - method construction
   - conditional (branching) statements
   - iteration = loops and methods
Major Course topics

2. Arrays:
   • collect data in arrays
   • searching
   • sorting
   • array manipulations
Major Course topics

3. Object Oriented Programming:
   - designing and constructing classes using containment
   - aggregation
   - inheritance
   - polymorphism
   - Application: GUIs
Instructor Information

• Dr. Paul Fodor
  214 New Computer Science Building

• **Office hours**: Tuesdays and Thursdays 3:30PM-5:00PM, online (on SBConnect).

• Email: paul (dot) fodor (at) stonybrook (dot) edu
  • Please include “CSE 114” in the email subject and your name in your email correspondence
General Information

- Course Web page: [http://www.cs.stonybrook.edu/~cse114](http://www.cs.stonybrook.edu/~cse114)
- **Blackboard** will be used for assignments, grades and course material.
General Information

• Class Time and Place (Summer 1: May 29, 2018 - Jul 7, 2018)
  • Summer calendar: http://www.stonybrook.edu/summer-session/calendar/
  • CSE 114-01 Computer Science I (Lecture 1): TuTh 12:00PM - 3:25PM, online (synchronous on SBConnect).
  • CSE 114-L01 Computer Science I (Laboratory): TuTh 3:30PM - 6:55PM, online (synchronous on SBConnect - teaching assistants: tbd).

• Exams (in person) at Stony Brook University:
  • Midterm Exam: Thursday, June 14, 2018, 12:00PM-2:00PM, during classtime, room Frey 104.
  • Final Exam: Thursday, July 5, 2018, 12:00PM-2:00PM, during class time, room Frey 104.
Textbook


- 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 9780134756370 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.
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:
  • Homework, project and labs = 30%
    • Programming homework assignments
    • Project
    • Lab assignments
  • Midterm exam = 35%
  • Final exam = 35%
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

- You will be given problems that require a programmed solution during lab hours and you must submit them in Blackboard or the textbook online programming system. The SBConnect teleconference system will also be available to contact the instructor and teaching assistants during the lab. At the end of the lab deadline, the instructor will assign grades of 0 - 3 as follows:
  - 0 - student did not attend lab
  - 1 - student attended lab and attempted to complete it, but it has major problems (doesn't compile or does nothing of value)
  - 2 - student attended lab and when executed, student work is partially complete
  - 3 - student attended lab and all work completed
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, and Elementary Programming</td>
</tr>
<tr>
<td>2</td>
<td>Selections, Mathematical Functions, Characters, and Strings, Loops and Methods</td>
</tr>
<tr>
<td>3</td>
<td>Arrays and Multi-dimensional Arrays</td>
</tr>
<tr>
<td>4</td>
<td>Objects and Classes, Object-Oriented Thinking, and Inheritance and Polymorphism, and Exception Handling and Text I/O</td>
</tr>
<tr>
<td>5</td>
<td>Abstract Classes and Interfaces</td>
</tr>
<tr>
<td>6</td>
<td>Recursion</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

• 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/ual/academicjudiciary](http://www.stonybrook.edu/ual/academicjudiciary)
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
- Java JDK standard edition:
  - http://www.oracle.com/technetwork/java/javase/downloads
- Eclipse IDE:
  - http://www.eclipse.org/downloads
- Learn to use the debugger!!
- Liang’s student Web site:
  - http://www.cs.armstrong.edu/liang/intro10e
Past CSE114 Projects: Spring 2012: Artificial Intelligence (AI) Poker
Fall 2012 CSE114 Project: AI Blackjack

User - purse=20; sum=19
Computer - sum=4

Round 1  Action: select bet

Bet:  $1  $2  Bet: $0

Action:  Stay  Hit

Result:  => New purse: $  Next

(c) Paul Fodor (CS Stony Brook)
Point 10

Good Luck

Roll Dices
Spring 2014 Project: Pai Gow Poker (double-hand poker)
Fall 2014 Project: 24 Game
Spring 2015 Project: Bridge

(c) Paul Fodor (CS Stony Brook)
Summer 2015 Project: Go Fish

(c) Paul Fodor (CS Stony Brook)
Fall 2015 Project: Wheel of Fortune
Spring 2016 Project: Cosmic Wimpout

http://cosmicwimpout.com/p/7/How-to-play
Fall 2016 Project: Scrimish
Spring 2017 Project: UNO!
Summer 2017 Project: the rush-hour game

http://www.thinkfun.com/play-online/rush-hour/

(c) Paul Fodor (CS Stony Brook)
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!