# CSCE 155A
## Introduction to Computer Science I
### Class Syllabus
#### Spring 2011

### Instructor

**Jeremy Suing**  
Computer Science Lecturer  
Office: 123C Kauffman or 368 Avery  
Phone: 402-472-1658  
Email: jsuing@cse.unl.edu  
Office Hours: 2:30 – 4:00 PM MF or By Appointment

### Teaching Assistants

<table>
<thead>
<tr>
<th>Shant Karakashian, Lab Manager</th>
<th>Abhishek Majumdar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office: Avery 123D</td>
<td>Office: Avery 123A</td>
</tr>
<tr>
<td>Phone: 402-472-4679</td>
<td>Phone: 402-472-4679</td>
</tr>
<tr>
<td>Email: <a href="mailto:shantk@cse.unl.edu">shantk@cse.unl.edu</a></td>
<td>Email: <a href="mailto:majumdar@cse.unl.edu">majumdar@cse.unl.edu</a></td>
</tr>
<tr>
<td>Office Hours: 9:00 – 11:00 AM R in CSE Resource Center</td>
<td>Office Hours: 10:30 AM – 12:30 PM R in CSE Resource Center</td>
</tr>
</tbody>
</table>

### Lectures & Labs

<table>
<thead>
<tr>
<th>Regular Lecture Section</th>
<th>Regular Lab Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time: 1:30 – 2:20 PM MWF</td>
<td>Section 151 3:30 – 5:20 PM T</td>
</tr>
<tr>
<td>Room 106, Avery Hall</td>
<td>Section 152 6:30 – 8:20 PM T</td>
</tr>
<tr>
<td></td>
<td>Section 153 10:30 – 12:20 PM W</td>
</tr>
<tr>
<td></td>
<td>Room 21, Avery Hall</td>
</tr>
</tbody>
</table>

### CSE Computing Facilities

This class is supported by the CSE Student Computing Laboratory, located in Avery 15. Students are provided an account for accessing laboratory computers hosting all software required for the course.

### CSE Student Resource Center & Other Resources

The [CSE Student Resource Center](#), located in Avery 13A, provides students in this class with first-line technical support, assistance with course work and assignments, and perspectives of upper-class CSE majors on academic and professional matters. The CSE Department [website](#) provides information about other departmental [resources](#).
Catalog Listing

(ACE 3)[ES] **CSCE 155A. Computer Science I** (4 credit hours) 3 hours Lecture, 2 hours Lab. Prerequisites: Appropriate score on the CSE Placement Exam or CSCE 101; MATH 103 or equivalent. This course is recommended for students majoring in computer science or computer engineering.

This class is an introduction to problem solving with computers. Topics include problem solving methods, problem analysis and specification, software development principles, algorithm development, program design, computer programming, implementation in a high-level programming language, and computing in society. Laboratory assignments develop mastery of a high-level programming language and programming practices.

**Prerequisites by Topic**

1. Appropriate score on the CSE Placement Exam
2. Mastery of basic mathematical problem solving as demonstrated by satisfactory completion of mathematics through college algebra, trigonometry, and pre-calculus (MATH103 or equivalent or appropriate scores on the Math Placement Exam)
3. Familiarity with the use of computers and software applications.

**Class Objectives**

The objective of this class is to familiarize students with problem solving approaches using computer programming. We study how to specify a problem, how to come up with a solution, how to convert the solution into a program, and how to solve the problem correctly using the program. Students learn how to analyze problems, develop requirements, design solutions, implement computer programs, and test software. Students will learn:

- a high-level programming language;
- elements of problem solving, including problem analysis, requirements and specifications, top-down design, algorithm development (including recursion), and step-wise refinement;
- software development principles, such as data and operation abstraction, encapsulation, modularity, reuse, exception handling, documentation, and testing;
- aspects of computing, such as algorithms to solve searching and other problems, graphical user interfaces, event-driven programming, and file access; and
- some history of computing.

The conversion of a solution to a program comes in many ways. In this class, we focus on Object-Oriented Programming (OOP) where we build “software objects” which are distinct and standalone that programs can use to carry out more complex tasks.
Text Book


An InterWrite Personal Response System (Required)

Relationship to ACE

This course will satisfy ACE Learning Outcome #3: Use computational and formal reasoning (including reasoning based on principles of logic) to solve problems, draw inferences, and determine reasonableness. Students have opportunities to acquire the knowledge and skills necessary to achieve the learning outcome in the lectures, homework and programming assignments and weekly laboratory assignments. The examinations, programming assignments, and laboratory assignments are used to assess achievement of the outcome. Additional details of the learning opportunities and outcome assessment are provided in the course specification.

Assignments

There will be several programming assignments to help you understand some of the basic concepts. The Java programming language must be used for all the programming assignments. The assignments are due at the beginning of class on the indicated dates.

You will also be required to hand in an on-line copy. In addition, you are required to hand in a hardcopy of your readme and testing files at class time on the due date. You are expected to write well-documented, modular code. Programs which do not compile or which do not run will not receive any credit. Programs that compile and run will be graded as follows:

- 45% Program Correctness,
- 15% Software Design,
- 10% Programming Style,
- 15% Testing, and
- 15% Documentation.

Personal Response System and Just-In-Time Quizzes

You are required use a personal response system (PRS) “clicker” that is registered in your own name in class for “just-in-time” quizzes and questions. Each quiz will have several questions, and will take a few minutes to complete. Anticipate having a set of such questions per lecture depending on in-class discussions and topics.
Examinations

We will have two mid-term (in-class) exams and a final exam. The dates for the tests are listed in the class schedule. There will be no makeup tests.

As part of CSE’s assessment of its academic programs, every student in this course is required to take a Progress Assessment Test (PAT) at the end of the semester. Sometime between Wednesday of the fourteenth week of classes (4/20/2011) and Wednesday of the fifteenth week of classes (4/27/2011), you will need to go into MapleTA in Blackboard to take an exam of 30-40 multiple choice questions based on content of this course.

Laboratories

You must register for a lab section as a part of this course. The laboratories are designed to supplement the lectures and provide hands-on experiences on topics that need additional attention. It is a significant part of the course and you are strongly encouraged to fully take advantage of this opportunity. You are required to read the lab handouts and the reading assignments before coming to the lab. You are also required to take the on-line pretest prior to coming to each lab. There are 15 laboratories altogether.

Final Grade

You will receive a numeric score for each programming assignment and exam. Each mid-term exam contributes 300 points, the PAT is worth 100 points, and the final exam is worth 500 points. The total for all required exams (two mid-term exams, PAT, and final exam) is 1200 points. The assignments contribute a total of 600 points. Each laboratory carries 40 points, totaling 600 points altogether. The PRS-based in-class quizzes contribute up to a total of 90 bonus points to the score. Class attendance (recorded via PRS) contributes up to a total of 30 bonus points to the score. As a matter of policy, no I (incomplete) grades will be given. Your final grade will be based on the following table.

<table>
<thead>
<tr>
<th>Points Range</th>
<th>Grade</th>
<th>Points Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 – 1440</td>
<td>F</td>
<td>1921 – 1960</td>
<td>B-</td>
</tr>
<tr>
<td>1441 – 1480</td>
<td>D-</td>
<td>1961 – 2080</td>
<td>B</td>
</tr>
<tr>
<td>1481 – 1600</td>
<td>D</td>
<td>2081 – 2160</td>
<td>B+</td>
</tr>
<tr>
<td>1601 – 1680</td>
<td>D+</td>
<td>2161 – 2200</td>
<td>A-</td>
</tr>
<tr>
<td>1681 – 1720</td>
<td>C-</td>
<td>2201 – 2320</td>
<td>A</td>
</tr>
<tr>
<td>1721 – 1840</td>
<td>C</td>
<td>2320 – 2400</td>
<td>A+</td>
</tr>
<tr>
<td>1841 – 1920</td>
<td>C+</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Course Schedule

Topics are introduced in readings, examined in lectures, practiced in laboratories, and demonstrated in assignments. The course schedule reflects the general ordering of topics. Students should complete assigned readings on topics before those lectures. Lectures on the topics will precede the corresponding laboratories and assignment due dates. Please refer to separate the “Class Schedule” document for details.

Disabilities

Students with disabilities are encouraged to contact the UNL Office of Services for Students with Disabilities (SSD) for a confidential discussion of their individual needs for academic accommodation. It is the policy of the University of Nebraska-Lincoln to provide flexible and individualized accommodation to students with documented disabilities that may affect their ability to fully participate in course activities or to meet course requirements. To receive accommodation services, students must be registered with SSD Office, 132 Canfield Administration, 472-3787 voice or TTY.

Academic Integrity

The CSE Department has a policy to promote and enforce academic integrity. Violations of academic integrity will result in automatic failure of the class and referral to the proper university officials. The work a student submits in a class is expected to be the student’s own work and must be work completed for that particular class and assignment. Students wishing to build on an old project or work on a similar topic in two classes must discuss this with both professors. Academic dishonesty includes: handing in another’s work or part of another’s work as your own, turning in one of your old papers for a current class, or turning in the same or similar paper for two different classes. Using notes or other study aids or otherwise obtaining another’s answers for an examination also represents a breach of academic integrity. Sanctions are applied whether the violation was intentional or not.

Those who share their code and those who copy other’s code will be penalized in the same way; both parties will be considered to have plagiarized.