Syllabus of CS115: Principles of Computer Science  
Fall 2009, Credit Hrs: 3, CS Dept/College of EAS

Time & Loc.: MW 3:05 PM -- 4:20 PM, ENG 107 (Lab sessions will be in the ENG 233)

Instructor:  
Prof. Joe X. Zhou  
Office: 176 Engineering Building, 255-3493 (office), Email: zbo@cs.uccs.edu  
Office Hours: MW 2:00 PM - 3:00 PM, 176 EAS, and/or by appointment (email preferred)  
Course website: http://www.cs.uccs.edu/~zbo/teaching/CS115/CS115_Fall09.html

Course Descriptions:

CS 115 gives the introduction to programming with emphasis on computer science concepts, particularly on the concepts of abstraction in problem solving. Students will learn to develop computer programs for problem solving, develop proficiency for programming in a modern programming language (Java), understand the basic principles of object-orientation and application to problem solving, and understand the selected advanced concepts in OO and Java. The course will also introduce basic concepts of computer systems and programming environments.

The specific course outcomes that the Computer Science Department has identified for this course are provided at the end of the syllabus. Most of them won't mean anything at this point, but by the end of the course you should be able to achieve all of the listed outcomes.

Course Textbook

The material presented in the course will be complemented by the following textbook.


Tentative Schedules

- Introduction (1 lecture)
- Primitive Data Types (2 lectures)
- Selection Statements (2 lectures)
- Loops (2-3 lectures)
- Methods (2 lectures)
- Arrays (3 lectures)
- Objects and Classes (3 lectures)
- Strings & Text I/O (2 lectures)
- Inheritance, Polymorphism, and ArrayList (3 lectures)
- Abstract Classes and Interfaces (2 lectures)
- Exception Handling and Object-Oriented Design (1-2 lecture)
- Lab sessions (6-7 lectures)

Prerequisites and Expectations

- high school algebra and familiarity with computer concepts including file operations and text editing.
- Although no prior programming experience is assumed, the pace of this course will be brisk since there is much material to cover. This course serves as one of the important cornerstones of the CS curriculum. Be prepared to work hard. The reward will be great!
Grading

The final grade will be composed of

- In-class discussion & Lab attendance 10%
- Homework programming assignments 35%
- In-class Quizzes (twice) 8%
- Midterm (in class, closed book and notes) 17%
- Final (in class, closed book and notes, comprehensive) 30%

All exams and quizzes will be carefully and fairly graded by the instructor himself. Grades will be assigned as follows:

- 90 ≤ {A}; 87 ≤ {A-} < 90; 84 ≤ {B+} < 87
- 80 ≤ {B} < 84; 75 ≤ {C+} < 80; 70 ≤ {C} < 75
- 65 ≤ {D+} < 70; 60 ≤ {D} < 65; F: below 60

Attendance

Class preparation is crucial to the success. Preview the text and bring the questions to the class.

Students should attend all lectures and labs. Missing lectures are extremely risky since the teaching pace is fast and the course is heavy (also in-class quizzes)! There will be concepts and examples presented in class that are not covered in the notes. Class discussion is always an important and useful component of each class. The last day to drop without special permission from your dean: Friday, Oct 30, 2009.

Computer Science Department Policy: “Students in 100 and 200 – level CS courses can have at most four absences, after which they would lose at least one letter grade.” Attendance is taken for all 100 and 200 level CS classes. Please contact me by email if you need to miss class and it will be handled case by case. Documentation may be required (e.g. a doctor’s note).

Cheating

Unfortunately, it is necessary to mention it here. Cheating on an exam or on homework may result in a grade of F in the course or expulsion from the University. Copying the work of another student whether that work is a homework program or an exam problem is cheating. Obtaining code via the Internet is cheating. You must write your own programs completely and not modify some other student’s work to disguise that the work has not originated from you. It is usually quite easy to see through such disguises. You are always welcome to discuss concepts with fellow students. You must draw a sharp line between discussing a concept and its implementation in a program. The former cooperation is allowed the latter is cheating. You are always welcome to chat with me about the design of a program. I shall take great care in clearing as many obstacles for you as possible but will not take away from you the challenge and learning value that you will derive from doing most of a problem yourself. Any work submitted for a grade must include the following statement and be signed and dated (as a hardcopy or e-version). If this is missing or not signed and dated, the work may be returned un-graded. Please prepare an envelope for your homework assignment submission.

We need the strict rules, because everyone wants to be, and will be, treated fairly in this class!

I have neither given nor received unauthorized assistance on this work.
Signed: Date:

Dropping course

The University gives you 10 weeks in which to drop any course. Please consult the official University calendar to determine the precise cutoff date. If you wish to drop the course before this cutoff date, I shall sign the necessary paper work to expedite this. After the cutoff date it is virtually impossible to drop the course without penalty unless the circumstances are severe and well documented. Then the Dean must approve such a drop. If you decide that you wish to drop the course, please do so before the 10 week deadline.
Getting help

Feel free to send me e-mail. I usually respond in a timely manner, but may not respond as quickly or at all on weekends or other break times.

You are very encouraged to see me and ask me questions during the office hours or by making appointments if outside the office hours. You are also welcome to drop by my office if I am available. I will be happy to point you in the right direction, clear obstacles and review the concepts associated with the particular problem but will not take away from you the challenge and learning value that you will derive from doing most of a problem yourself.

Homework assignment policy

Each student must accomplish the assigned homework problems individually. While you may discuss the concepts and algorithms before developing the program solutions, the solutions and the code, which you hand in, must be written by yourself. **You may not work together on the actual coding of the assignments.** The turning in of duplicate (or near duplicate) code will be seen as cheating and will result in a zero grade for all students involved.

Homework assignment submission requirement

Each homework assignment may contain a set of problems with a due date specified usually one week from the date the problem set is assigned. Assignments will be posted on my website and handed out in class. Each problem in an assignment shall receive a separate grade.

Each homework assignment should be turned in at the beginning of class on the due date. In case you cannot complete a problem by the beginning of class on the due date, you can take another two days to work on the problem and turn it. The penalty for such a period late submission will be 20 percent. If you hand the problem in two class periods late, the penalty shall be 40 percent for that problem. Beyond one week from the specified due date the problem shall NOT be graded for any reason.

Please assemble all homework in an envelope or folder of your choice. I shall not accept loose homework. The folder should keep the contents from falling out and contain:
1. A clear header including your name (and your partner name in case some lab assignments), assignment number and problem number.
2. A copy of the homework assignment sheet.
3. A printout of all source code and supporting comments.
4. A printout of the output from each program (you may use Alt+PrtSc to copy the console output).
5. A 3 ½ inch DOS-format floppy (or a CD-ROM, or use a USB drive) containing all source code needed to compile and run your program. This diskette must not contain any files unrelated to the problem set. I shall compile and run each program that you submit as well as examine your source code. This source code must be nicely and consistently formatted. Unformatted or difficult to decipher code shall result in a grade reduction for the particular problem being graded.
6. If your problem is one or two class periods late you must clearly state this on top of the first sheet.
7. A self-assessment of each problem. This should indicate whether you believe you have completed the problem successfully. It may also discuss any special difficulties that you have had in solving the problem.

Programs will be graded by compiling and running them on a PC configured like the lab computers. Make sure that the programs can be tested at the DOS console, by the use of `javac` and `java` commands. I will NOT use any IDE to grade the programs, though you can use any IDE such as NetBeans to develop the programs. Please verify the contents of your disk or CD before turning it in. It has not been uncommon to receive disks that contain nothing. That is the grade that is awarded.

If a program does not compile at the DOS console, it shall receive an automatic grade of 0. If a program produces run-time or logical errors you shall receive only partial credit.
Exams

There will be about 2 class quizzes, which will be taken in the class. There will be one midterm exam and one final exam, which are close-book and close-notes. The midterm exam will be in the class, Monday, Oct 12, 2009, ENG 107. The final will be during 1:40PM - 4:10 PM, Wednesday, Dec 16, 2009, ENG 107. There will not be any makeup exam! The exception may be granted to extreme non-academic situations, such as illness (evidence required).

Pair programming in the labs

There's lots of evidence in industry that programming in pairs (called pair programming) leads to higher-quality software. In addition, numerous professors have found that pair programming can lead to better student learning. The Computer Science Department has decided to use pair programming in all our CS 115 lab assignments this semester to see how well that approach works here at UCCS.

For each lab assignment, you'll work in pairs to complete the assignment. Please note that working in pairs in the labs does NOT extend to the homework programming assignments, where such an action would be cheating; pair programming is only for the lab assignments completed during class/lab time.

Others

If you have a disability for which you are requesting an accommodation, you are encouraged to contact the Disability Services Office, located in Main Hall #105 (Phone # 255-3354), within the first week of classes.

CS 115 COURSE OUTCOMES

- Understands the basics of computer problem solving and how implemented in the chosen language (currently Java).
  - Understands that all computer problem solving has two parts: data and operations.
  - Understands that data must be declared (type and identifier) and initialized before use. Understands the difference between scalar and reference types.
  - Understands the mechanics of operation definition and use.
  - Understands the mechanics of installation and use of the chosen language (currently this means installation and setup of Java on a Windows based PC plus editing, compiling and running Java programs with and without the use of an IDE such as NetBeans).

- Is capable of solving problems using fundamental types and operations.
  - String and integer types, newly defined types as new classes.
  - Understands conditional, iterative and branching control constructs.
  - Understands how to do simple input/output.

- Understands the basic principles of object-orientation and application to problem solving.
  - Understands the defining OO concepts: Object, Class, Instance, Message, Method. Understands how to define and use classes and objects.
  - Understands the essential OO features: abstraction, encapsulation, inheritance, polymorphism.
  - Understands that OO programming consists of sending messages to objects (objects are data, messages invoke operations defined by methods).
  - Understands that static fields and operations are rarely justifiable. Understands the meanings of modifiers for class, field and method.
  - Understands the operations defined for all objects (in class Object) and the need to redefine selected inherited operations.

- Understands selected advanced concepts in OO and Java.
  - Understands definition/rules/purpose/use of an interface and that it is also a type representing a precise but common behavior or property.
  - Understands simple data structures (arrays and ArrayList) and how to create/use them.
  - Understands the concept of an Iterator and how to use it to access objects in data structures.
  - Has experience building/using classes in an application requiring the use of polymorphic substitution and dynamic binding for both sub-classes (inheritance) and classes that implement interfaces.
<table>
<thead>
<tr>
<th>Week#</th>
<th>Class#</th>
<th>Date</th>
<th>Lesson/Activity</th>
<th>Reading</th>
<th>Homework</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>M 8/24</td>
<td>Introduction, Lab#1: Hello Java program + JDK + NetBeans</td>
<td>Chapters 1, 2</td>
<td>Supplements</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>W 8/26</td>
<td>Primitive data types &amp; operations</td>
<td>Chapter 2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>M 8/31</td>
<td>Primitive data types &amp; operations</td>
<td>Chapters 2, 3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>W 9/2</td>
<td>Lab#2</td>
<td>Chapter 3</td>
<td>HW1 assigned</td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>M 9/7</td>
<td>No class (Labor day)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>W 9/9</td>
<td>Selection Statements</td>
<td>Chapters 3, 4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>M 9/14</td>
<td>Selection Statements, Loops</td>
<td>Chapter 4</td>
<td>HW1 due</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>W 9/16</td>
<td>Loops</td>
<td>Chapter 4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>M 9/21</td>
<td>Loops, Lab#3</td>
<td>Chapters 4, 5</td>
<td>HW2 assigned</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>W 9/23</td>
<td>Methods</td>
<td>Chapter 5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>M 9/28</td>
<td>Methods, Lab#4</td>
<td>Chapters 5, 6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>W 9/30</td>
<td>Arrays</td>
<td>Chapter 6</td>
<td>HW2 due</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>M 10/5</td>
<td>Arrays</td>
<td>Chapter 6</td>
<td>HW3 assigned</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
<td>W 10/7</td>
<td>Lab#5</td>
<td>Chapters 1-6</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td>M 10/12</td>
<td>Midterm</td>
<td>Chapter 6</td>
<td>HW3a due</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>W 10/14</td>
<td>2D Arrays</td>
<td>Chapters 6, 7</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>16</td>
<td>M 10/19</td>
<td>Objects and Classes</td>
<td>Chapter 7</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>17</td>
<td>W 10/21</td>
<td>Objects and Classes</td>
<td>Chapters 7</td>
<td>HW3b due</td>
</tr>
<tr>
<td>10</td>
<td>18</td>
<td>M 10/26</td>
<td>Objects and Classes</td>
<td>Chapter 7</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>19</td>
<td>W 10/28</td>
<td>Strings and Text I/O</td>
<td>Chapter 8</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>M 11/2</td>
<td>Strings and Text I/O</td>
<td>Chapter 8</td>
<td>HW4 assigned</td>
</tr>
<tr>
<td>11</td>
<td>21</td>
<td>W 11/4</td>
<td>Lab#6</td>
<td>Chapters 8, 9</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>22</td>
<td>M 11/9</td>
<td>Inheritance and polymorphism</td>
<td>Chapters 9</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>23</td>
<td>W 11/11</td>
<td>Inheritance and polymorphism</td>
<td>Chapter 9</td>
<td>HW4 due</td>
</tr>
<tr>
<td>13</td>
<td>24</td>
<td>M 11/16</td>
<td>ArrayList</td>
<td>Chapter 9</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>25</td>
<td>W 11/18</td>
<td>Lab#7</td>
<td>Chapters 9, 10</td>
<td>HW5 assigned</td>
</tr>
<tr>
<td>14</td>
<td>26</td>
<td>M 11/23</td>
<td>Abstract Class &amp; Interfaces</td>
<td>Chapter 10</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>H</td>
<td>W 11/25</td>
<td>Thanksgiving (no class)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>27</td>
<td>M 11/30</td>
<td>Attending IEEE Conf. (no class)</td>
<td>Chapter 10</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>28</td>
<td>W 12/2</td>
<td>Lab#8</td>
<td>Chapter 10</td>
<td>HW5 due</td>
</tr>
<tr>
<td>16</td>
<td>29</td>
<td>M 12/7</td>
<td>Abstract Class &amp; Interfaces</td>
<td>Chapter 10</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>30</td>
<td>W 12/9</td>
<td>Exception handling; wrap-up review for refinal</td>
<td>Chapters 1-11, 18.1-5</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>31</td>
<td>W 12/16</td>
<td>Final Exam</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* There might be some changes due to unpredictable conditions, such as weather, conference trip, etc. If any change, it will be announced in the class and posted in the course Web site.

* Labs will be in ENG 233