CSIS 3701: Advanced Object-Oriented Programming

Instructor:
Dr. John R. Sullins
Office hours:
  Monday 10:00 – 11:00, 2:00 – 2:45, 4:30 – 5:00
  Wednesday 10:00 – 11:00, 2:00 – 2:45, 4:30 – 5:00
  Friday 10:00 – 10:30
  Or by appointment
Office: 307A Meshel Hall
Phone: 330-941-1806
Email: jrsullins@ysu.edu
Twitter: @jrsullins3701
Web site: http://cis.ysu.edu/~john/3701

Check the web site regularly, as assignments and announcements will be posted here.

Objectives:
1. Introduce the principles of object-oriented programming (classes, objects, messages, encapsulation, static properties, inheritance, polymorphism, exception handling, and object-oriented containers).
2. Introduce core sections of the Java language related to object-oriented programming.
3. Apply object-oriented programming to the design, development, and testing of large programs.

Note that this course is not meant as a comprehensive introduction to all of Java -- the primary objective is to use Java to introduce concepts of object-oriented programming, design and development.

Prerequisite:
CSIS 2610: Programming and Problem Solving or CSIS 2605: Fundamentals of Programming and Problem Solving.

It is assumed that you are familiar with C/C++ syntax (since Java is heavily based on it), and that you can design and implement reasonably large programs involving those concepts.

Textbook:

Programming Assignments:
The programming assignments will initially consist of a couple of shorter exercise to introduce you to the Java language, and then move on to larger project covering major concepts of object-oriented programming.

The Java SDK (software development kit) and the NetBeans IDE are available on the lab PCs, and you are strongly encouraged to install a copy on your own PC from the Java web site (instructions for this are on the course web page).

Work on these assignments must be your own. See below for more details.
Grading:

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Percentage</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming assignments</td>
<td>50%</td>
<td>7 or 8 exercises and projects</td>
</tr>
<tr>
<td>Exam 1</td>
<td>12%</td>
<td>9/29</td>
</tr>
<tr>
<td>Exam 2</td>
<td>12%</td>
<td>10/20</td>
</tr>
<tr>
<td>Exam 3</td>
<td>12%</td>
<td>11/17</td>
</tr>
<tr>
<td>Final Exam</td>
<td>14%</td>
<td>Wednesday, 12/8, 8am – 10am</td>
</tr>
</tbody>
</table>

Last day to withdraw with a "W": Saturday, October 25.

Tentative Course Outline:

<table>
<thead>
<tr>
<th>WEEK</th>
<th>TOPICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/20</td>
<td>Introduction to object-oriented design and the NetBeans IDE</td>
</tr>
<tr>
<td></td>
<td>(no class Monday)</td>
</tr>
<tr>
<td>8/25</td>
<td>Introduction to Java: Basic syntax, types, and memory management</td>
</tr>
<tr>
<td>9/1</td>
<td>Using object and classes; Strings, I/O, and other built-in Java classes</td>
</tr>
<tr>
<td></td>
<td>(no class Monday)</td>
</tr>
<tr>
<td>9/8</td>
<td>Static class methods; Application structure</td>
</tr>
<tr>
<td>9/15</td>
<td>Creating object-oriented support classes; Basic Java class syntax</td>
</tr>
<tr>
<td>9/22</td>
<td>Introduction to object-oriented class design and unit testing</td>
</tr>
<tr>
<td>9/29</td>
<td>Encapsulation; GUI and event classes</td>
</tr>
<tr>
<td></td>
<td>(Exam 1 Monday)</td>
</tr>
<tr>
<td>10/6</td>
<td>Visual applications and layouts</td>
</tr>
<tr>
<td>10/13</td>
<td>Exception handling and validation</td>
</tr>
<tr>
<td>10/20</td>
<td>Object-oriented design and decomposition; Universal Modelling Language</td>
</tr>
<tr>
<td></td>
<td>(Exam 2 Monday)</td>
</tr>
<tr>
<td>10/27</td>
<td>Inheritance and Polymorphism</td>
</tr>
<tr>
<td>11/3</td>
<td>The Java object hierarchy; Design for inheritance</td>
</tr>
<tr>
<td>11/10</td>
<td>Generalized containers and iterators</td>
</tr>
<tr>
<td>11/17</td>
<td>Interfaces; Callbacks and Java GUI event handling</td>
</tr>
<tr>
<td></td>
<td>(Exam 3 Monday)</td>
</tr>
<tr>
<td>11/24</td>
<td>Streams and file manipulation</td>
</tr>
<tr>
<td></td>
<td>(no class Wednesday or Friday)</td>
</tr>
<tr>
<td>12/1</td>
<td>Advanced topics: Database manipulation in Java, JSP for web design, etc.</td>
</tr>
<tr>
<td>12/8</td>
<td>Final Exam (Wednesday 8:00am – 10:00am)</td>
</tr>
</tbody>
</table>
General Course Policies and Guidelines:

**Grading:**
The course grade will be based on the required material:

- 90% will guarantee at least an “A” for the course,
- 80% will guarantee at least a “B” for the course,
- 70% will guarantee at least a “C” for the course, and
- 60% will guarantee at least a “D” for the course.

**Due Dates and Late Assignments:**
An assignment (including programs and projects) is late if it is not IN MY POSSESSION (either as hardcopy or electronically) at the beginning of class on the due date. Late assignments may be penalized at some percentage (usually 10%) per day late, and no credit will be given for assignments turned in after solutions have been handed out. Extenuating circumstances (such as nonfunctional labs) may be recognized if they become a chronic problem.

For a programming course like this, *keeping up with the assignments is particularly important*, as current topics tend to build off of the previous topics (in other words, if you skip an assignment, you will probably still have to learn the material for the next assignment anyway). In addition, the best way to study for an exam in a programming course is to have written the related programs.

**Attendance:**
Class attendance is optional, except for students who are receiving VA benefits, or in situations (such as group meetings) where your absence would be detrimental to other students in a group.

However, missing class is *not* an acceptable excuse for failure to complete required material on time. Every lecture will cover material related to assignments and exams, and in general the grades in programming classes are directly related to the number of lectures attended. Material that is presented in class will not be covered again outside of class -- if you miss class, it is up to you to find out what was covered and to get the notes from someone else. In general, *success in this class is related to attendance in lectures and labs*.

**Exams:**
Exams will cover material presented in class and corresponding required sections in the text, and will also usually relate to material covered in the homework. Makeup exams are allowed, but only for compelling and verifiable reasons. I need to be informed as soon as possible if you need to take a makeup (ideally, *before* the exam is given), and I reserve the right to refuse if too much time has passed since the exam, or if no compelling reason is given.

**Office Hours:**
The best way to get help with an assignment is to stop by my office during office hours. Many problems that you might get “stuck” on for hours can be fixed with my help within a couple of minutes. The earlier that you come to me with a question, the more likely you are to finish the assignment on time, so come to me for help early.
If you are having problems with a program, be sure to bring a copy of the program (either on your laptop, a flash drive, or a printout).

**Email:**
The best way to reach me with questions outside of office hours is email (john@cis.ysu.edu). I will attempt to answer within 48 hours (except for holidays, weekends, and breaks). There are some things that you can do to help out:

- Include your **name** and **course number** in the subject (otherwise it might not make it through the spam filters).
- Be as specific as possible about the question or problem.
- If it is a problem with a program, be sure to attach it. However, depending on the type of program and where I happen to be, I may not be able to help right away (office hours are usually better for getting help with programs).

Most assignments that you write for the class will be submitted via email. When submitting assignments by email, do the same things:

- Include your **name**, **course number**, and the **number of the assignment** in the subject.
- Attach all code/documents. If there are issues with the size/number of attachments, please use gzip to compress into a single file.

**Academic Honesty:**
Academic honesty is expected and required. HELPING fellow students is acceptable, and is actually a very good way to learn the material (particularly with debugging programs). COPYING is NOT acceptable, and will result in loss of credit for the assignment, and possibly failure of the course. Follow these guidelines:

- All work on the **design** and **basic coding** phase of a program should be your own. That is, sitting in a group writing a program or assignment together is considered to be copying (unless this is specifically allowed as part of a group project).
- If you receive help with debugging part of an assignment, then you must acknowledge that help in the documentation of that section (your grade will not be affected unless otherwise announced).
- If you give help to another student, then it is your responsibility to make sure that they fully understand the problem and solution -- just giving someone code is worse than no help at all.
- Under NO circumstances should code be copied from one student's file to another (it's a lot easier to detect than you might think).

The bottom line: if you are not sure how to approach a problem, or are stuck at some point, **SEE ME FIRST FOR HELP**.

Unless specified otherwise, all exams are **closed book** (this includes notes, phones, etc.). Any suspected cheating on an exam will result in failure for the course.
For further information, see the section on Academic Dishonesty in the Undergraduate Bulletin. See also the CSIS Acceptable Use Policy for Lab Standards at http://www.cis.ysu.edu/Meshel.Hall.Labs/aup.html.

### Classroom Etiquette:
Your fellow students deserve an environment without disruptions to learning. Examples include:

- Talking during lecture.
- Printing in labs during lectures.
- Texting/tweeting.
- Web surfing.
- Cell phone use.
- Arriving at class later/leaving class early.
- Eating or drinking (prohibited in our labs).

On the other hand, *asking questions* during lecture is very strongly *encouraged*. If you are confused about a topic, chances are that many other people are as well!

### Incomplete Grades:
Incomplete grades are strongly discouraged. However, an incomplete grade may be assigned under the following conditions:

- The student must request in writing that an incomplete grade be assigned.
- The student's previous work in the course must have been satisfactory.
- The reason(s) must be beyond the student's control, and deemed justifiable by the instructor.

Insufficient time is NOT a justification for an incomplete. Also note that all incompletes must be made up within two months; otherwise, they automatically revert to an F.

### Disability Services:
In accordance with University procedures, if you have a documented disability and require accommodations to obtain equal access in this course, please contact me privately to discuss your specific needs. You must be registered with the Center for Student Progress Disability Services, located at 275 Fifth Avenue, and provide a letter of accommodation to verify your eligibility. You can reach CSP Disability Services at 330-941-1372.

### Academic Support:
The Marion G. Resh Center for Student Progress is a resource on Campus established to help students successfully complete their university experience. Please phone (330) 941-3538 or visit the Center for assistance in tutoring or for individualized assistance with social and academic success. The main Center is located in Kilcawley West below the bookstore.