Syllabus for Introduction to Programming with Java

Vital Statistics:
- **When:** TR 1:00 – 2:20 PM
- **Where:** 222 Technology Center
- **Who:** Prof. Amanda Holland-Minkley
  ahollandminkley@washjeff.edu
  228 Technology Center
  724-503-1001 ext. 3400
  Office Hours: MW 1:00-2:00, R 11:00-12:00 (or by appointment)
- **Computer Lab:** 204 Technology Center
- **Student Assistant:** Tyler Dykins, dykinstd@washjeff.edu
  Help Sessions to be announced

Important Dates:
- First Day of Class: Wednesday, February 1st
- Last Day to Add/Drop: Tuesday, February 7th, 5:00 PM
- First Exam: Thursday, March 2nd
- Last Day for S/U or Withdrawal: Monday, March 27th
- Second Exam: Thursday, April 6th
- Last Day of Class: Tuesday, May 9th
- Final Exam: Friday May 12th, 3:00 – 5:30 PM

Catalog Course Description:
An introduction to object-oriented programming, with an emphasis on program design, problem solving, methodology, and algorithms. Students will master the fundamental concepts and structures common to programming.

Course Objectives:
By the end of the semester, you should be able to:
- Create a complete Java program from scratch as well as modify an existing program
- Understand the process of algorithm design
- Select and use different data types and control structures as appropriate in context
- Write readable, well-organized, well-commented code
- Locate and fix bugs in Java programs

Required Text:
*Java Software Solutions*, 4th edition, Lewis & Loftus, ©2005 Addison Wesley. – Be sure that you have the copy of the text with the Maple leaf on the cover indicating that the book covers Java 5.0!

Students are not required to have or use their own computer for the course, as all software needed will be available in Rooms 222 and 204 of the Technology Center as well as the open lab on the first floor. However, for those students interested in using their own computers, all software being used is open source and available for free download over the internet; further information on software packages will be distributed as they are introduced in the class.

Assignments, Evaluation and Grading:
There will be four primary forms of evaluation in this class: in-class activities, short homeworks, major projects, and exams.

© 2006, Amanda M. Holland-Minkley. This document may not be reproduced or distributed, in whole or in part, without the author’s permission.
A significant portion of class time will be used writing code or doing other in-class activities, including short in-class quizzes. Your efforts on these activities will be worth 10% of your final grade. Due to the nature of the activities, you will not generally be able to make up this work outside of class without documented extenuating circumstances and, if possible, prior agreement of the instructor. Note that full credit on these exercises may not require perfectly running code so much as a demonstration of active engagement with the code.

Short homework assignments will be due most weeks, with there being eight to ten homework assignments across the entire semester. These assignments will test specific skills covered in class during the previous week. These short homeworks will be worth 30% of your final grade. While the short homeworks will usually involve writing code, they may also incorporate written exercises in code design or interpretation.

There will also be three longer programming projects assigned in which you will be expected to write longer pieces of code and incorporate many concepts and skills. These projects will each be worth 10% of your final grade.

Finally, the remaining 30% of the final grade will be based on two exams during the semester and a comprehensive final exam, each worth 10% of the final grade. The two midterm exams will be held in class on Thursday, March 2nd and Thursday, April 6th. The final exam for the course will be held Friday May 12th from 3:00 – 5:30 PM.

Late Policy:
All assignments must be turned in promptly. Late assignments will be accepted, but at a 5% reduction in grade for each day that they are late (defined as each 24-hour time period after the original due date). All assignments, unless otherwise noted, will be submitted through the Moodle course management system. It is your responsibility to ensure that your assignment has been submitted correctly and in a timely fashion. A detailed description of the format to be used in submitting assignments will be distributed with the first homework assignment; some assignments may come with additional formatting requirements which you will be expected to follow.

Academic Honesty Policy:
It is assumed that you have read and are familiar with the college’s Academic Honesty Policy. Applying the policy to this course, it is expected that all work submitted will be yours alone, unless you are explicitly permitted to work with others on that specific assignment. In the context of programming, using another’s code, whether from a book, internet source, or other student, without attribution of the source is considered plagiarism. In addition, using another’s algorithm, or general plan for a program, is also considered plagiarism. Both are a violation of the academic honesty policy. All cases of academic misconduct will be pursued per the procedures laid out in the academic honesty policy and may result in failure of the class and reporting to the Office of Academic Affairs.

In general, what this means is that assignments must be done independently and without reference to another student’s work. Any outside sources used in completing an assignment, including internet references, must be fully cited on any homework assignment or exercise. The exception to this is code or algorithmic suggestions given to you by either the instructor or the student assistant, which need not be cited (though it would not be in error to do so). Reuse of your own code from previous assignments is also permitted, though for significant lengths of code reuse, you should indicate its original source.

Exams will be held to a similar standard – all code produced must be your own work. Exams will be open book, but you will not be permitted access to code you have previously written or other repositories. Any
violation of these or other rules governing exams set out when they are administered will result in failure of that exam and possibly the course.

**General Course Mechanics and Classroom Expectations:**

Though this course does not have lab sections, a significant portion of regular class meetings will be run in a lab-like fashion, so you need to come to class prepared to participate. It is required that you bring your text, paper and a writing utensil, and some form of removable media to every class session with you, including the exams. These materials will not be provided for you. You are expected to attend class regularly, and you are fully responsible for any readings or assignments given in class, in-class activities, and for content covered in class which is not in the course text.

Note that W&J does not recognize “excused absences”. However, we do acknowledge that attendance is sometimes impossible due to emergency or other extenuating circumstances. These situations can only be accommodated if you make arrangements with the instructor, in advance if possible, or at the earliest possible time after the missed class (do not wait until the next class meeting). In the case of an emergency that causes you to miss meetings of multiple courses, you should contact the Dean of Academic Affairs or Dean of Student Life; they will be able to contact your instructors and help you make any necessary arrangements.

*It is impossible to learn to program without actually practicing programming*, and in order to do well in this course, you must plan on programming frequently between class meetings. It will be important that you do all of the readings promptly as assigned, as lectures will assume you have done so. Given the nature of this course, you will not pass if you do not come to class. Furthermore, learning programming is a cumulative process. Everything builds on previous material, so if you do not understand something, do not wait for the next topic to come along. Get help from either the instructor or the student assistant promptly.

We will be making use of the Moodle course management system available at http://www.washjeff.edu/moodle/; be sure that you can log in and access the course pages through it and contact me if you have any problems. You are responsible for checking the system regularly for updates about assignments or class meetings, particularly if you miss class. You should also check your e-mail on a regular basis; I will assume that all students in the class are reachable through their W&J e-mail account.

Help sessions will be held by the student assistant regularly (probably two evenings a week), in room 204 of the Technology Center. The student assistant will be able to review class content and give assistance on assignments, but will not write code for you. Learning to program for many is a process of trial-and-error, and learning to diagnose and fix errors is an important skill to learn, so while we may give you suggestions as to how to proceed, you will be expected to do the work yourself. This may mean that you have to ask for assistance many times in the course of writing a program. As with office hours, plan to bring your course text and be ready to run and debug your code in the lab during help sessions.

In addition, students may use the lab outside of help sessions to work on course assignments or practice programming. All course software will be installed and configured on the computers in the lab to match the equipment in room 222. The lab is unlocked weekdays until 5:00 PM. After 5:00 PM on weekdays, and all day on Saturday and Sunday, the lab will be accessible to students enrolled in the course by keycard. At 10:00 PM, security will check the labs and students will only be permitted to remain if there is more than one individual in the lab; at 2:00 AM security will do a final sweep of the labs and all remaining students will be asked to leave. This policy is also posted on the door of the lab; if you have any difficulties with lab access, please let me know.
A final note on class meetings: As part of our in-class programming exercises, I will frequently be looking at your code and projecting student’s screens to the front of the room for everybody to see. This is a good time to consider what your background image is, and to ensure that no personal information is visible on your screen during class. Further, be aware that even if your screen is not being currently projected, I can still see what is happening on it from the front of the room; when working in room 222 do not assume any privacy or security. Finally, you may not use the computers for non-class related activities during the class session. It is distracting to myself and to your fellow students. If disruptions become a problem, I reserve the right to lock or shut down student computers without notice.

**Topics:**
The following is a rough calendar of the topics that we will be covering in class. Planned content and project due dates may be schedule to revision.

<table>
<thead>
<tr>
<th>Week</th>
<th>Dates</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>2/2</td>
<td>Introduction</td>
</tr>
<tr>
<td>Week 2</td>
<td>2/7, 2/9</td>
<td>Data Types &amp; Expressions</td>
</tr>
<tr>
<td>Week 3</td>
<td>2/14, 2/16</td>
<td>Data Types &amp; Expressions</td>
</tr>
<tr>
<td>Week 4</td>
<td>2/21, 2/23</td>
<td>Using Classes</td>
</tr>
<tr>
<td>Week 5</td>
<td>2/28, 3/2</td>
<td>Exam 1, Using Classes</td>
</tr>
<tr>
<td>Week 6</td>
<td>3/7, 3/9</td>
<td>Writing Classes</td>
</tr>
<tr>
<td>Week 7</td>
<td>3/14, 3/16</td>
<td>Project 1 due, Writing Classes</td>
</tr>
<tr>
<td>Week 8</td>
<td>3/28, 3/30</td>
<td>Conditionals</td>
</tr>
<tr>
<td>Week 9</td>
<td>4/4, 4/6</td>
<td>Exam 2, Loops</td>
</tr>
<tr>
<td>Week 10</td>
<td>4/11, 4/13</td>
<td>Loops, Project 2 due</td>
</tr>
<tr>
<td>Week 11</td>
<td>4/18, 4/20</td>
<td>Object-Oriented Design</td>
</tr>
<tr>
<td>Week 12</td>
<td>4/25, 4/27</td>
<td>Arrays</td>
</tr>
<tr>
<td>Week 13</td>
<td>5/2, 5/4</td>
<td>Arrays</td>
</tr>
<tr>
<td>Week 14</td>
<td>5/9</td>
<td>Wrap-up, Project 3 due</td>
</tr>
</tbody>
</table>
Guide to Classroom Responsibilities

**Students’ Responsibilities:**
- Come to class, on time and prepared.
- Submit all work in a timely fashion, in the format required.
- Abide by the college’s Academic Honesty Policy.
- Be attentive and participate in class discussions.
- Do not be distracting or disruptive to others in the class.
- Read all course documents and messages from the instructor carefully.

Most importantly, be aware of what you do and do not understand. Start assignments early; look at returned assignments and solution sets promptly and follow up on any problems. Ask questions often and early – if you are confused about something, odds are other people are too. Go to office hours – that’s what they are for – or make an appointment for an alternate time. However, please do bring your notes, assignment, text or whatever material for which you have questions along with you. Communicate with me – I am here to help you learn the material, but you have to speak up if there is a problem.

Take an active role in your learning!

**Instructor’s Responsibilities:**
- Be on time and prepared for class.
- Grade and return all work in a timely fashion, giving clear feedback.
- Enforce the college’s Academic Honesty Policy consistently.
- Attempt to find effective ways to communicate course content to all students in the class.
- Be open to student questions during class and encourage participation.
- Ensure that all students have a comfortable environment in which to learn.
- Communicate all expectations and requirements to students clearly and in a timely fashion.
- Treat all students respectfully and equitably.

My responsibility is to give every student the best possible opportunity to master course material and meet the stated course objectives. Understanding that every student comes to a class with different interests and motivations, and that these needs must be balanced within the classroom, it is my goal to create a classroom experience in which all students are supported as well as challenged.

It is my responsibility to create an environment in which you can actively learn.