

CS A 120 Computer Science I

Michael Goldwasser
Spring 2004

Dept. of Mathematics and
Mathematical Computer Science
Saint Louis University

Contents

1	Overview	2
2	Course Administration	2
2.1	The People	2
2.2	Class Meetings	3
2.3	Textbooks	3
3	Practice Problems	3
4	Graded Work	3
4.1	Programming Assignments (50%)	3
4.2	In-Class Quizzes (20%)	3
4.3	Exams (30%)	4
4.4	Course Grades	4
4.5	Academic Integrity	4
4.6	Late Policies	5
5	Students with Documented Disabilities	5
6	Online Resources	6
6.1	CS A 120 Web Page: http://euler.slu.edu/~goldwasser/120/	6
6.2	The Textbook's Website	6
6.3	Java Documentation	6
6.4	BlueJ, our chosen development platform	6
6.5	Electronic Assignment Submission	6

1 Overview

Computer Science I is an undergraduate course which provides an introduction to computer programming. No prerequisite programming experience is required, though we will expect a sufficient preparedness in analytical thinking and abstraction, as equivalent to four years of high school mathematics.

What distinguishes computers from many other types of machines and inventions, is their great generality and their programmability. Personally, one can find great joy and pride in instructing a computer to behave precisely as desired; there can also be great frustration at times, when finding that the instructions given to the computer were not actually interpreted as intended. We will hope to lead you through a great many successes during this course; at the same time, it is almost certain that your path will lead you through some times of frustration and even some failed attempts. Keep your spirits up when tackling the challenges and rely on the instructor for help.

This semester, there is particular excitement. Our department has been working on a new rendition of this course to go into effect for all sections starting in Fall 2004. We will be serving as a pilot group for this newly revised version. The major curricular change to the course is that we will be teaching a useful style of programming known as *Object-Oriented Programming*.

Modern pieces of software are often large and complex. To design such systems, you cannot easily rely on the same techniques which you might have used when writing small programs. Instead large-scale software design involves recognizing distinct “components” or “objects”, understanding how these objects must interact with each other. Only then should you worry about the low-level implementation of these components.

In step with this change of paradigm, we will be changing the programming language used for instruction, choosing to use the object-oriented *Java programming language*, rather than the imperative *C programming language*. (Note: If any student was specifically aiming for learning the C programming language, there is another section offered this semester, CS A 120-01, which will be using C as the language).

2 Course Administration

2.1 The People

Instructor: Dr. Michael Goldwasser
Email: goldwamh (-at-) slu.edu
Web: <http://euler.slu.edu/~goldwasser/>
Office: Ritter Hall 6
Phone: (314) 977-7039

Office hours: Mondays, 10–11am
 Tuesdays, 12–1pm
 Fridays, 10:30–11:30am
 or by appointment

Please make sure to take advantage of office hours, as they offer a wonderful opportunity for individual attention. Though some of the class periods will be run in a lab setting, with time for individual help, the unavoidable fact remains that such a class period allows for an average of only 4 minutes of individual attention per student.

2.2 Class Meetings

The lectures

The material will be presented in three weekly lectures.

Time: Mon/Wed/Fri, 12:00–12:50pm

Place: Shannon Hall 118

Class participation is most welcome. The schedule and copies of lecture notes can be found on the course schedule web page.

2.3 Textbooks

The required textbook for this class is

Title: *Java, Java, Java: Object-Oriented Problem Solving, Second Edition*

Author: Ralph Morelli

Copyright Date: 2003

ISBN: 0-13-033370-0, Prentice Hall

Author's Website: <http://starbase.trincoll.edu/~jjjava/>

Publisher's Website: http://wps.prenhall.com/esm_morelli_jjjoops_2

3 Practice Problems

To reinforce conceptual material from the class, we will often suggest a set of practice problems to consider. These will not be submitted and thus you may discuss them freely with each other and with me. At times, we will go over these problems in class. The timing of these practice problems will often be coordinated as preparation for quizzes and exams.

4 Graded Work

4.1 Programming Assignments (50%)

We expect there to be a total of 6-8 programming assignments during the course. On certain assignments, you will be required to work individually; on others you will be allowed to work in pairs. Unless otherwise specified, the programming assignments will be weighted equally. At the end of the semester, we will drop your lowest score and will use the remaining scores to contribute to this portion of your course grade.

Note that half of your course grade is being determined by these assignments. This reflects the great amount of time and effort which you will be putting forth in completing these assignments. When completing this work, please respect the policy on Academic Integrity as given below.

Programming assignments will be submitted electronically; details will be announced later.

4.2 In-Class Quizzes (20%)

There will be six quizzes during the course, general every other Friday, each lasting approximately 15–20 minutes. The precise schedule for quizzes will be found on the course schedule web page. At the end of the semester, we will drop your lowest score and will use the remaining scores to contribute to this portion of your course grade.

We strongly suggest that you use the provided practice problem as an aide in preparing for these quizzes.

4.3 Exams (30%)

There will be two in-class exams, one at mid-semester and one at the end of the semester.

- Midterm Exam (10%), Friday, 5 March 2004, 12:00–12:50pm
- Final Exam (20%), Wednesday, 5 May 2004, 12:00–1:50pm

4.4 Course Grades

Letter grades will be based on each student's overall percentage of awarded points according to the above contributions.

- Student percentage above 90% will result in a grade of A or better.
- Student percentage above 85% will result in a grade of B+ or better.
- Student percentage above 80% will result in a grade of B or better.
- Student percentage above 75% will result in a grade of C+ or better.
- Student percentage above 70% will result in a grade of C or better.
- Student percentage above 65% will result in a grade of D+ or better.
- Student percentage above 60% will result in a grade of D or better.
- Student percentage below 60% will result in a grade of F.

Any modification to this scale at the end of the course will be *in favor of the students*. That is we may later decide to award an A to a student who is slightly below the above cutoff, but we certainly will not take away an A from someone who is just above the cutoff.

4.5 Academic Integrity

Students are expected to have read and abide by the University statement on Academic Integrity available on page 56 of the Saint Louis University's Undergraduate Catalog. The College of Arts & Science provides a more detailed policy statement which applies within the College and thus to this course. This can be found at <http://www.slu.edu/colleges/AS/academichonesty.html>

When it comes to learning and understanding the **general course material**, you may certainly use whatever outside references you wish, and may have discussion with other students of this class or other people from outside the class. This openness pertains to material from the text, practice problems, general syntax and use of the Java language or other computing tools.

However, when it comes to **work which is submitted for this course**, any such use of unauthorized sources is prohibited. You are free to have consultations with the instructor, teaching assistants, or members of organized tutoring centers on campus. Even in these cases, if you receive significant help you should make sure to document both the source of the help as well as the extent. Furthermore, we will explicitly allow students to work in pairs on certain assignments, in which case conversations between partners is justified (in fact necessary).

Other than those sources, you are in no way to discuss such assignments, nor are you to use or search for direct or indirect assistance from any outside references, including but not limited to:

- other students in this class
- past students, whether from this school or other schools
- other acquaintances
- other texts or books
- online information not referenced on the course or text web page

Any violations of these policies will be dealt with seriously. Penalties will apply as well to a student who is aiding another student. Any such violations will result in a minimum penalty of a zero on the given assignment which cannot be dropped, and severe or repeated violations will result in an immediate failing grade in the course. Furthermore all incidents will be reported in writing to the Department and/or the Dean, as per the College procedure.

4.6 Late Policies

The timing of all components of the course are designed in a coordinated way. A student will be best served by staying current on all facets of the course.

- All **exams and quizzes** must be taken promptly at the required time. Requests for rescheduling an exam or quiz will only be considered if the request is made in advance, or else in an “emergency” situation with appropriate documentation.
- For **programming assignments**, we wish to allow students to continue to work comfortably beyond an “official” deadline, in case more time may result in more progress. At the same time, we wish to discourage students from falling too far behind pace and jeopardizing their success on future assignments.

Our solution is the following exponentially decaying late formula with a half-life of approximately five days (some have suggested that we should offer extra credit to anyone who fully understands this formula).

We will consider an assignment submission “complete” when any part of the assignment is last submitted or modified. Any assignment which is not complete promptly by its due date and time will be assessed a penalty based on the formula $S = R \cdot e^{-h/173}$, where S is the grade given, R is the grade the work would have received had it been turned in on time, and h is the amount of time (in hours or fractions thereof) that the work was late.

Examples:

- work turned in 1 hour late will receive 99.6% of its original credit
- work turned in 5 hours late will receive over 97% credit
- work turned in one full day late receives less than 88% credit
- work turned in two full days late receives less than 76% credit
- work turned in five days late will receive under 50%.

The above policies will be waived only in an “emergency” situation with appropriate documentation.

5 Students with Documented Disabilities

In accordance with the Americans with Disabilities Act, reasonable accommodations may be made to assist a student with a documented disability.

Such student should immediately contact the Disabilities Services Office in Suite 130 of the Academic Resources Center or at (314) 977-2930. The Disabilities Services Office can assist you and the instructor in formulating a reasonable accommodations plan and provide support in developing appropriate accommodations for your disability. Course requirements will not be waived, but accommodations may be made to assist you to meet the requirements. Technical support may also be available to meet your specific need.

6 Online Resources

This course will take advantage of the Internet and the campus network in many ways.

6.1 CS A 120 Web Page: <http://euler.slu.edu/~goldwasser/120/>

With the exception of the first week's printed handouts, most of the information for this course will be distributed only by means of the course web page. This web site will contain all assignments, a schedule of lectures, detailed lecture notes and links to many other sources of information.

The web page contains some information (e.g. solutions, submitted assignments, individual grades) which is more sensitive and therefore which will be available to students in the class only after they have identified themselves properly.

6.2 The Textbook's Website

- Publisher's Official Website
<http://vig.prenhall.com/catalog/academic/product/0,4096,0130333700,00.html>
Not much of interest, but links to the other sites
- Companion Website
http://wps.prenhall.com/esm_morelli_jjjoops_2
This is organized chapter by chapter, and for each chapter has a summary of Objectives, a self-study quiz, and Power Point slides provided by the author. In addition, from this home page you can find a link which is a ZIP archive containing all source code from the text.
- Author's Companion Website
<http://starbase.trincoll.edu/~jjjava/>
Also contains complete source-code from text (available as single archive, or individual files). This site also contains demonstrations of working software for various "CyberPet" examples from the text.

6.3 Java Documentation

A good collection of Java documentation links has been compiled by our textbook's author, at <http://starbase.trincoll.edu/~jjjava/links.html>

6.4 BlueJ, our chosen development platform

BlueJ is a **freely available**, integrated Java environment specifically designed for introductory teaching. It was developed and is supported by several Universities.

Its home page, at <http://www.bluej.org>, contains further information, including a download if you wish to install it on your own computer.

6.5 Electronic Assignment Submission

All assignments for this course must be submitted electronically. Details of the procedure will be announced at a later time.