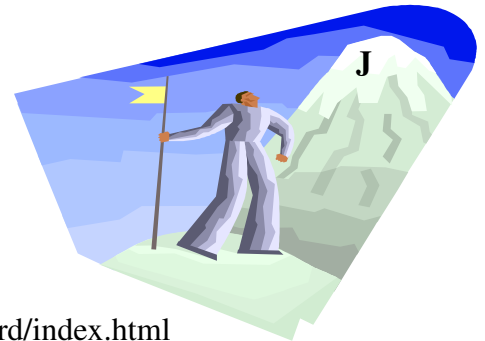


**Computer Science Department
Loyola College**

CS 202 - Computer Science II

Instructor: Roberta E. Sabin
Office: Donnelly Science 127C
Telephone: (410) 617-2562
FAX : (410) 617-2157
E-mail: res@loyola.edu



Black Board Course Website: <http://www.loyola.edu/blackboard/index.html>

Visit the CS Department WebSite: <http://www.cs.loyola.edu>

Class Meets: MWF 10-10:50 (in KH006), R 9:25-10:40 (in DS121)

Office Hours: Monday, Wednesday, Friday: 11-11:50
Thursday: 10:50-12:15

Others times by appointment. (I am usually here every weekday except Tuesday.)

One must learn by doing the thing, for though you think you know it, you have no certainty until you try,

--Sophocles, 5th century B.C.

In programming, it is not enough to be inventive and ingenious. One also needs to be disciplined and controlled in order not to become entangled in one's own complexities.

--Harlan D. Mills, Forward to Programming Proverbs by Henry Ledgard

The price to be paid for reliability is simplicity.
Hoare

-- C. A. R.

...for heaven's sake, place as much weight on useful intelligent comments as on the elegance of the algorithms!...we have some programs of 5-15 million lines of code that are barely commented and ...are unmaintainable...So we have to build around them when we upgrade our systems, not knowing what might break if we change them

--A Bell Systems' Programmer by E-mail

Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform.

--Ada Augusta

Byron, 1843

CS 202

Computer Science II

4 credits

SYLLABUS

Catalog Description:

Prerequisite: CS201.

A continuation of CS201. Emphasizes structured programming skills and introduces more advanced programming features such as object design and reuse, recursion, and simple data structures using a high-level, object-oriented language.

Texts: Koffman and Wolz, *Problem Solving with Java*, Addison Wesley, 2002 (2nd edition).
Eastman, Roger, Lecture Notes (unpublished manuscript).

Supplemental Software: On the CD included with the text and at

http://wps.aw.com/wps/media/access/Pearson_Default/483/495288/login.html

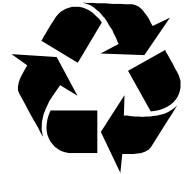
Java documentation: <http://www.mycore.de/library/jdk1.2.2/docs/api/overview-summary.html>

jGrasp documentation: <http://www.eng.auburn.edu/grasp/>

Course Objectives:

Upon successful completion of this course, the student should be able to:

1. understand to some degree the major theoretical principles underlying computer science
2. understand the major hardware and software components of a computer system
3. understand and use object-oriented design in software development and represent class relationships with simple UML diagrams
4. write, test, and debug programs that utilize any and all constructs available in standard Java
5. have some understanding of object-oriented programming and the use of classes in Java
6. be able to create a simple GUI using Java constructs
7. understand and be able to use sorting, searching and recursive algorithms
8. have some understanding of time efficiency issues as they relate to choices of algorithms
9. enunciate and intelligently respond to ethical issues related to computing
10. understand and implement elementary data structures (list, stack, queue) using Java



What you can expect of me: You can expect that I will come to class prepared and ready to help you learn. You can expect me to be enthusiastic (easy since I LOVE computer science and teaching!), be knowledgeable, and keep the class moving. You can expect me to be available during my office hours and at other times that you arrange to see me. Expect me to return graded work promptly. You can expect me to treat you respectfully.

What I expect of you: I expect you to come to class prepared to contribute to class—computing is an active sport. You CANNOT learn it in the passive mode. This means that you should have completed the assignment, done the reading, and determined what you need help in understanding. You contribute to class by intelligently questioning the instructor and offering further explanation to me and your colleagues. I expect you to take responsibility for learning computer science—you won't be sorry. Further, I expect you to treat every other member of the class (including me) with respect.

Course Outline of Class Lectures: Notice that each Monday and Friday class will normally concentrate on programming in Java. On Wednesday, the lecture will be largely devoted to a “breadth-first” CS topic and preparation for Thursday’s lab. CS Topics are denoted in *italics*. Thursday labs (not shown in the schedule) are normally devoted to discussion of programming assignments and exercise that will facilitate your completion of the programming assignments.

Class No.	Date	Major Topics	References ¹
-----------	------	--------------	-------------------------

¹ Numerals indicate text sections; E indicates the Eastman text

² B indicates a breadth-first topic

1	1/12 M	Introduction to the course Arrays (one dimensional) and sorting	Syllabus 5.1-5.4
2	1/14W	<i>Computer codes</i> Discussion of Lab 1: Processing an integer array	<i>E6.1</i>
L	1/15R	Lab 1: Processing an array of integers	
3	1/16 F	Arrays of objects Multi-dimensional arrays .	5.5-5.6
4	1/21W	<i>How information is coded</i> Discussion of PA1	<i>E6.2</i>
L	1/22 R	Lab 2: PA1 work	
5	1/23 F	Vectors and “wrapping”	5.7-5.8
6	1/26 M	Class hierarchies and inheritance PA1 due	6.1-6.2
7	1/28 W	<i>Error-correction, encryption, and compression</i> External files	<i>E6.3</i> 9.1
L	1/27R	Lab 3: Using external files Quiz 1	
8	1/30 F	Polymorphism UML diagrams	6.3
9	2/2 M	Interfaces and sorting revisited	6.4
10	2/4 W	<i>Computer Hardware Models</i>	<i>E7.1</i>
L	2/5 R	Lab 4: PA2 work	
11	2/6 F	Exceptions Files in Java	8.1 8.2
12	2/9 M	Using text files	8.3
13	2/11 W	<i>Hardware: CPU</i> PA2 due	<i>E7.2</i>
L	2/12 R	Lab 5: CPU Simulator and Discussion of PA3	
14	2/13 F	Binary files Quiz 2	8.5
15	2/16 M	GUIs and event-driven programming Java GUI components	7.1-7.2
16	2/18 W	Creating a GUI	7.3
L	2/19 R	Lab 6: a baby GUI	
17	2/20 F	<i>Hardware: Circuits</i>	<i>E7.3</i>
18	2/23 M	Retrofitting a GUI to an application	7.4
19	2/25 W	More GUI components	7.5-7.8
L	2/26 R	Test 1	
20	2/27 F	GUI wrap-up and final PA questions	
<p>The class schedule for the second half of the semester will be distributed on 3/8. The second half of the course will cover elementary data structures, some GUI construction, and miscellaneous additional topics.</p>			
<p>Exam: Monday, May 3, 1 PM, KH006</p>			

GRADING

Your final grade will be determined by the number of points that you earn. Approximately 1000 points will be possible this semester. They will result from:

Approximately 10 collected assignments	and 4 quizzes	=	250	points
6 programming assignments (approximately)		=	350	points
2 tests (100 points each)		=	200	points
1 final exam		=	200	points
	Total	=	1000	points

Final letter grades will be calculated on a scale close to the following:

A = 91 - 100% (905-1000 points)	C = 71 - 74% (705-744 points)
A- = 88 - 90% (875-904 points)	C- = 68 - 70% (675-704 points)
B+ = 85 - 87% (845-874 points)	D+ = 65 - 67% (645-674 points)
B = 81 - 84% (805-844 points)	D = 60 - 64% (595-644 points)
B- = 78 - 80% (775-795 points)	F = 0 - 59% (594 or fewer points)
C+ = 75 - 77% (745-774 points)	

1. Tests and quizzes are announced. Consult the class schedule.
2. Normally, a homework assignment is due at the beginning of each class. If an assignment is collected, it is assigned a value of 10-20 points. Generally, half the points are awarded for completeness and half for correctness. Correctness is determined by my checking all or a portion of the assigned exercises.
3. Programming assignments will be given periodically. Generally 10 days to 2 weeks are provided for the completion of a programming assignment. Late programming assignments are NOT accepted, unless so stated when the assignment is given. Programming assignments are due at the beginning of the regular class period. See class schedule for dates.
4. Please keep a record of the points you have earned. This will enable you to calculate your current average and correct errors on my part.

CLASS PROCEDURES

0. This is a 4 credit course. You should expect this course to claim more time and effort than a 3 credit course. Your GPA will be affected accordingly. (You should expect to invest at least $4 \times 3 = 12$ hours per week outside of class time on this course.)
1. You are expected to have read the reference material listed in the class schedule before the lecture. Both the textbook and the "breadth-first" notes present material in a very concise manner. You will greatly enhance your experience of the lectures by coming to class prepared. I normally distribute many handouts--copies of notes, assignment sheets, etc. In the past, many students have increased their success in this course by keeping notes, programs, tests, and assignments organized. A three-ring binder is recommended (sections for breadth-first topics, class notes, home assignments, and programming assignments may be helpful.)
2. I will use **Blackboard** for dissemination of class materials. Blackboard will also hold copies of code snippets, classnotes, programming assignments, weekly assignments, data files, and important announcements. Normally, you will submit your programming assignment BOTH in hardcopy and as a file send to me via the Blackboard DropBox.
3. Regular attendance is necessary for success in this course. As a point of courtesy, plan to arrive on time for class. Students are responsible for material presented and assignments made during absences. Normally, make-up exams are not administered and LATE ASSIGNMENTS ARE NOT ACCEPTED.
4. This is a "hands-on" course. Therefore, besides reading the texts, attending all classes, taking good notes, completing assignments and studying, to be successful in this course, you will need to spend a substantial amount of time in the lab or at your computer. Many students find they need five to eight such hours per week. Try to avoid pulling "all-nighters" to complete programming assignments. No one thinks clearly or uses time effectively when he or she is exhausted. Start planning a programming assignment when it is first distributed. Remember that the Intro Programming Lab (DS121) is usually quiet and not crowded.
5. Should you need extra help, please see me during office hours or make an appointment for a mutually convenient time. Do NOT wait until you are totally lost and have failed a test or quiz. Your success is my first priority. Even if I am busy or seem

distracted, please know that I am determined to be available to you for your assistance.

6. Assignments that are not printouts should be submitted on standard size paper. Please make them legible and neat. Illegible and/or sloppy assignments will NOT be graded.
7. All assignments are to be completed by the student as an individual unless otherwise noted. Collaboration on algorithm design is encouraged. However, sharing code or answers to exercises is NOT allowed. (See below.)
8. Academic honesty is required of all Loyola students at all times. It is expected that all students abide by the Honor Code:

The Honor Code states that all students of the Loyola Community have been equally entrusted by their peers to conduct themselves honestly on all academic assignments.

The students of this College understand that having collective and individual responsibility for the ethical welfare of their peers exemplifies a commitment to the community. Students who submit materials that are the products of their own minds demonstrate respect for themselves and the community in which they study.

All outside resources or information should be clearly acknowledged. If there is any doubt or question regarding the use and documentation of outside sources for academic assignments, your instructor should be consulted. Any violations of the Honor Code will be handled by the Honor Council

Any copying of an assignment, whether electronically or by hand is considered plagiarism. Students submitting non-trivial projects with identical structure will be considered to have acted dishonestly. Such students may be referred to the Honor Council for disciplinary action. At the very least, two or more students presenting assignments identical in all important aspects will share the points from a single grade. Discussion among students on how to attack programs is encouraged. However, **as soon as code is exchanged, the line between collaboration and plagiarism has been crossed.**

You will be asked to sign the following pledge that will appear at the bottom of all tests and quizzes:

I hereby declare that I have abided by the Honor Code during this test.

9. If you have a disability of any type that requires an accommodation, please let me know as soon as possible.