

CS291 – Syllabus for Spring 2006

Important Note:

Depending on your background, this course might require significant work on your part. In addition to the lectures, you should plan for about **8-12 hours of your own time for course work each week.**

Course Goals:

This course will give you an introduction to computer programming. Besides learning the basics of the programming language C++, this course has a heavy focus on problem solving. You will learn and practice to analyze and understand given problems and then design, implement and test solutions for them.

This course will introduce you to a methodology of systematic problems solving used in computer science and related fields. This approach to problems will be useful in many different areas beyond computing or computer science.

Lecture:

Mondays, 1:00-2:50 PM in AC 2B04
Wednesdays, 1:00 – 2:50 PM in AC 2B04

Instructor:

Detlef Ronneburger

Email: ronnebg@york.cuny.edu

Office: AC 2C07c

Office Hour: Mondays, 3:00-4:00 and 5:30 – 6:30 PM
Wednesdays, 12:00-1:00 PM
(and by appointment)

Phone: 718-262-2545 (Mondays and Wednesdays)
732-810-1614 (Tuesdays and Wednesdays)

Course Web Page

www.york.cuny.edu/~ronnebg/CS291.html

Required Course Work:

- Regular attendance is required (see below for attendance policy)
- For each lecture a ½ page write-up (typed), summarizing what you learned in that lecture (see details below)
- Homework: weekly programming assignments and review questions
- Programming project that includes several incremental steps throughout the semester
- One Midterm
- One Final

Textbook:

Computing Concepts with C++ Essentials, 3rd Edition
by Cay Horstman
Publisher: John Wiley & Sons, ISBN 0-417-16437-2
THIS TEXTBOOK IS REQUIRED!

- Grading Policy:**
- 15 % Class Participation, Attendance, Lecture Summaries
 - 15 % Homework (written and programming combined)
 - 20 % Course Project
 - 20 % Midterm
 - 30 % Final
 - 0 % Begging
 - 0 % Bribing

NOTE: You will need a passing grade both on the final as well as on the programming project(s).

Exams: The exams will be open book, open notes exams.

Homework Assignments: There will be short weekly programming assignment as well as written review questions. These assignments will be fairly easy, but they are important to ensure that your programming experience grows continually. There will not be detailed grading on these assignment, depending on your work you will receive full credit, half credit or no credit for each assignment. The homework assignments will have to be solved by YOU ALONE, not with the study group.

Programming Project There will be a larger programming project that you will work on during the entire semester. The work on this project will be split up into several smaller steps that you will have to complete by their respective deadlines. If you have not solved the previous step, you will not be able to move on to the next. THE WORK ON THE PROJECT WILL BE PERFORMED IN STUDY GROUPS OF 2 PEOPLE.(See *Study Groups* below)

Lecture Summaries For each lecture you will have to write up a summary (200-400 words) explaining in your own words what you learned during that lecture. This write-up needs to be submitted on paper, AT THE BEGINNING of the next lecture.

Study Groups The instructor will assign you to a study group of fellow students. Each group will have two students. You will solve the different parts of the programming project as part of your study group. You will receive a group grade as well as and individual grade for the project. EVERY GROUP MEMBER HAS TO CONTRIBUTE EQUALLY. If a group member does not contribute, or takes over all the work, it will affect his/her individual grade negatively.

Late Submissions

DEADLINES are STRICT. Unless you have made prior arrangements with the instructor, you will be penalized for late submissions. The penalties are as follows

Number of times late	On time	1 day late	2 days late	3-4 days late	> 4 day late
First time	-0%	-0%	-10%	-20%	-33%
Second time	-0%	-10%	-20%	-33%	-50%
3 rd -5 th time	-0%	-20%	-33%	-50%	-66%
> 5 times	-0%	-33%	-50%	-66%	-80%

Attendance Policy:

Class attendance and participation is MANDATORY. It will contribute 15% to your overall grade. So even with perfect scores on all assignments, if you have a spotty attendance record, you will probably get a grade lower than you would with a good attendance record.

If you have special circumstances that might interfere with your regular attendance (work responsibilities, child care responsibilities, etc.), please discuss these with your instructor.

Two health related absence per semester will be accepted without documentation. For any other unavoidable absence, documentation is required. If you have any foreseeable reasons to be absent, the instructor needs to be informed as soon as you become aware of the situation. Failure to do so in a timely manner will cause the absence to be counted as unexcused.

Academic Integrity:

ALL WORK SUBMITTED MUST BE YOUR OWN! Submitted programs will be checked for cheating using the anti-plagiarism software MOSS. You are allowed to and even encouraged, to discuss general ideas with fellow students, but you are not allowed to share anything written. You have to explicitly state on your submission, who if anybody you discussed the problem with. You are allowed to ask other people for help, if your program does not compile properly – BUT THIS HELP IS RESTRICTED TO CORRECTING YOUR PROGRAM SO **THAT IT WILL COMPILE AND NOTHING ELSE**. You have to acknowledge such help on your submission. The instructor reserves the right to enforce the college guidelines for handling plagiarism to their strictest interpretation.

Course Topics

The following course outline indicates the individual topics and sections of the book we plan to cover each week. Please note that this is a preliminary list. The up-to-date version will be available on the blackboard course web-page.

CS 291 – Preliminary Course Outline

Week	Topics Covered	Sections
1	<ul style="list-style-type: none"> • Review: <ul style="list-style-type: none"> ○ Vectors ○ Strings ○ If/Else, While, For ○ Variables: global vs. local ○ Functions: pass by value, pass by reference ○ C++ Programming Environment 	
2	<ul style="list-style-type: none"> • Objects and Classes <ul style="list-style-type: none"> ○ Review: Using Objects ○ Combining Related Data and Functions 	6.1 – 6.4
3	<ul style="list-style-type: none"> • Objects and Classes <ul style="list-style-type: none"> ○ Constructors ○ Data access and protections 	6.5 – 6.8
4	<ul style="list-style-type: none"> • Pointers: <ul style="list-style-type: none"> ○ Dynamic Memory Allocation 	10.1 – 10.2
5	<ul style="list-style-type: none"> • Pointers: <ul style="list-style-type: none"> ○ Pointers to Objects 	10.3
6	<ul style="list-style-type: none"> • Pointers and Arrays <ul style="list-style-type: none"> ○ Array vs. Vectors ○ 2D arrays ○ Arrays vs. Pointers ○ Characters Arrays 	4.4 – 4.7
7	<ul style="list-style-type: none"> • Catch-up, Review and Midterm 	
8	<ul style="list-style-type: none"> • Linked List: <ul style="list-style-type: none"> ○ Using Linked List provide by C++ Standard Template Library ○ Using iterators 	16.1
9	<ul style="list-style-type: none"> • Linked List: <ul style="list-style-type: none"> ○ Implementation ○ Variations: Double Linked Lists and Circular Lists 	16.2
10	<ul style="list-style-type: none"> • Stack & Queue <ul style="list-style-type: none"> ○ Using STL provided Stacks and Queues ○ Implementing them as descendants of Lists: Inheritance 	16.3
11	<ul style="list-style-type: none"> • Recursion <ul style="list-style-type: none"> ○ Recursively Implementing List Algorithms ○ Mutual Recursion 	14.3 – 14.5
12	<ul style="list-style-type: none"> • Sorting and Searching: <ul style="list-style-type: none"> ○ Selection Sort, Insertion Sort ○ Merge Sort 	15.1 – 15.5
13	<ul style="list-style-type: none"> • Sorting and Searching: <ul style="list-style-type: none"> ○ Binary Search 	15.6 – 15.8
14	<ul style="list-style-type: none"> • Catch-Up, and Review 	