

1. Instructor

John Arras
4122 A.V. Williams Bldg.
jra@cs.umd.edu

2. Teaching Assistant

Gutemberg Bezerra

3. Description

The topics covered by this course are the following: file I/O, dynamic memory allocation, object-based programming, recursion, searching algorithms, sorting algorithms, and data abstraction. You will learn the essentials of the C++ Programming language required to study the above topics.

4. Pre/Co requisites

CMSC 106 (Introduction to C Programming), Corequisite: MATH 141

5. Text

Text: *C++: How to Program*, Deitel & Deitel, Third Edition, Prentice Hall

6. Posting Accounts

Course materials (e.g., projects, assignments, office hours, etc.) will be made available in the public course posting accounts. Directions for obtaining course materials from the posting accounts will be discussed in class.

7. Office Hours

The Instructors' and TAs' office hours will be 12:30-2:00 every day after discussion and lecture and will be held in 1151 AV Williams. You can also make an appointment to see me.

8. E-mails

The Instructors' emails are supplied so you can contact us when necessary in emergency situations, but we regret that we have insufficient resources to debug students' projects via email. If you have any problems with your projects then stop by during the instructors' office hours or the TAs' office hours. Office hours or personal communication during or after class (if time permits), are the preferred means for asking us questions about the course material. Given the extra lecture time added into the course, there should be some time for project questions during lecture.

9. Course Requirements

Quizzes (announced):	Several	10%
Midterms:	two midterms	30% (15% of the total course grade each)
Programming projects:	seven coding assignments	35%
	Project #1 – 4%	
	Project #2 – 4%	
	Project #3 – 4%	
	Project #4 – 4%	
	Project #5 – 8%	
	Project #6 (Required)– 8%	
	Project #7 – 3%	
Final exam:	will be comprehensive	25%

The final course grades may be curved, if necessary, based on each student's total numeric score for all coursework at the end of the semester.

You must provide a working version of your Project #6 in order to pass the course. A working version of a project is one that satisfies the primary output as specified in the project description. **ANY STUDENT WITHOUT A WORKING VERSION OF PROJECT #6 WILL RECEIVE AN F IN THE COURSE.** For example, if a student has a 100 in every project, exam, and quiz, but does not have a working version of Project #6, then the student will receive an F in the course.

10. Grading Policy

All projects will be due at the time and date indicated on each project assignment. All projects will be graded out of 100 points. After the deadline for a project, you have two extra days to submit a working version of your project. Only the last project submission you provide will be graded; it will supercede any earlier submissions you may have made. For example, if a student has both an on-time as well as a two-day late submission then only the last (two-day late) submission will be graded. The penalties for each day late are as follows:

1 day late -> -10 points (you can get a maximum of 90 points)
2 days late -> -20 points (you can get a maximum of 80 points)

If after the third day you have not provided a working version of your project (except for Project #6) then you will receive 0 points for the project. Project #6 is the only exception to this rule. We will accept Project #6 submissions after the second day late, and until August 5 at 4:00 p.m. Versions of Project #6 submitted after second day late and before August 5 at 4:00 pm will receive a maximum of 10 points.

Under **no circumstances** will any emailed project, or a project submitted by any means other than that specified on the assignment be considered.

11. Missed Assignments

There will be no make-ups for missed hourly exams or quizzes. In the case of a documented medical excuse or documented family or personal emergency the grade for a missed exam/quiz will be calculated by averaging the student's scores for other exams and/or quizzes. Please note that "Verification of Visit" forms from the University Health Center do not constitute a valid excuse – an authorization form to release medical information must be signed for any Health Center documentation to be considered valid (this form is provided by the Health Center.) If it is known in advance that a student will miss an assignment, the student must notify the instructor prior to the due date of the assignment.

12. Important Dates

Exams will be held on the dates below unless a change is necessary to accommodate lecture topics. In that case the revised date will be announced in lecture. The final exam date, however, is fixed. IF YOU HAVE A SITUATION THAT PREVENTS YOU FROM TAKING THE FINAL EXAM AT THE SCHEDULED DATE IS YOUR OBLIGATION TO INFORM YOUR INSTRUCTOR DURING THE FIRST TWO WEEKS OF CLASS, IN ORDER TO MAKE THE APPROPRIATE ARRANGEMENTS.

Exam 1: Wednesday, June 23

Exam 2: Wednesday, July 21

Final Exam: Friday, August 6

Submission of Project #6 working version: Thursday August 5 at 4:00 pm.

13. Learning Assistance Service

If you are experiencing difficulties in keeping up with the academic demands of this course, contact the Learning Assistance Service, 2201 Shoemaker Building, 301-314-7693. Their educational counselors can help with time management, reading, note-taking and exam preparation skills.

14. Academic Integrity Statement

Many students in this course believe that we will not check that each project has been implemented individually. We want to inform you that we do check. We have software tools that allow us to compare every student project against every project across all the sections of the course. Some students believe that by changing variable names, adding/removing comments, etc., they can transform one program into a different one; this is not true. Even with those modifications we are able to recognize one project as a modified version of another project and we will report this as an academic integrity situation. Our advice to you is to do all your projects by yourself.

Every semester we have taught we have caught students cheating on project assignments. Student's academic careers have been ruined by their decision to cheat.

Please carefully read and consider the Computer Science Department policy regarding the use of computer systems, and the Academic Instructional Technology Services' policy regarding acceptable use of computer accounts provided for instructional use, in the Fall 2001 issue of the departmental newsletter, CS Tid-Bits, handed out with this syllabus. Note in particular the penalties for impermissible cooperation on programming projects, which is a violation of the University's Code of Academic Integrity. **Any evidence of unacceptable use of computer accounts or unauthorized cooperation on exams or projects will be submitted to the appropriate authorities or to the Student Honor Council, which could result in an XF for the course, suspension, or expulsion from the University.**

NOTE SPECIFICALLY that projects are to be done INDIVIDUALLY. **For academic honesty purposes, projects are to be considered comparable to an extended take-home exam. That is, any cooperation or exchange of ideas which would be prohibited on an exam is also prohibited on a project assignment.** Working IN ANY WAY with another student will be considered a violation and WILL BE REPORTED to the Honor Council. Students may not discuss pseudocode, design of a program or of individual functions, coding of individual functions, or any high-level code with **anyone** except the instructors and teaching assistants. Should you have difficulty with a programming assignment you should **see your instructor or the teaching assistants in office hours**, NOT solicit help from anyone else in violation of these rules. Students are welcome to study together or to receive help in learning the course concepts from any others, but quizzes, exams and projects are to be solely and entirely each student's own work.

VIOLATIONS OF ACADEMIC HONESTY INCLUDE:

- failing to do all or any of the work on a project by yourself, other than assistance from the instructional staff.
- using any ideas or any part of another student's project, or copying any other individual's work in any way.
- giving any parts or ideas from your project, including test data, to another student.
- having programs on an open account or on a PC that other students can access.
- transferring any part of a project to or from another student or individual by any means, electronic or otherwise.

IT IS THE RESPONSIBILITY, UNDER THE UNIVERSITY'S HONOR POLICY, OF ANY STUDENT WHO LEARNS OF AN INCIDENT OF ACADEMIC DISHONESTY TO REPORT IT TO THEIR INSTRUCTOR.

15. Accommodations for Students with Disabilities

Students with disabilities must inform the instructor of their needs at the beginning of the semester.

16. Course Topics (SUBJECT TO CHANGE)

This list may vary according to the pace of lecture topics.

	Topics
Week 1	Introduction, cin/cout, control structures, types, casting, operators, functions, arrays, structures, basic I/O
Week 2	More I/O, iosflags, Stream Manipulators, Stream Error States, Namespaces, Null Pointer, Strings, BNF, File I/O
Week 3	Passing Streams as Arguments, open/close, Moving through files, Effective Software Development, Program organization, Makefiles, Const with Pointers and References, Passing Arguments, Pointer Arithmetic
Week 4	Arrays of Pointers, Dynamic Memory Allocation, Dynamically allocated Arrays, Member Functions, Classes, String Class
Week 5	Static Members, Copy Constructors, "This", Classes with dynamically allocated members, "free" and "copy", Const in Parameter List, Temporary Objects, Const Objects and Const Member functions
Week 6	Recursion, Static Const, Objects inside Objects, Arrays of Objects inside Objects, Mutable
Week 7	Member Initializer Lists, Conversion Constructors, Linked Lists
Week 8	Recursion with Linked Lists, Friends, Operator Overloading, Abstract vs. Concrete
Week 9	Searching, Big-O Notation, Sorting, Forward Declarations, Proxy Classes, Exception Handling
Week 10	More Exception Handling, Singleton Class, Overloading new and delete, Hashing