

CMPSC 101.4 Introduction to Algorithmic Processes

Winter 1977

MWF 5th

62 Willard

General Information

Instructor: Donald Braffitt
320 Whitmore Lab
865-1545
Office Hours:

Assistant: Jeff Bross
321 Whitmore
865-7292
Office Hours:

Grader:

Text: Richard Conway and David Gries - Primer on Structured Programming

Exams: Midterm - Wednesday, January 19, 1977

Final -

<u>Grading:</u>	7 Quizzes @ 5% each	35%
	Midterm Exam	15%
	Final Exam	20%
	8 Assignments @ 3% each	24%
	Class Participation	6%
		<hr/>
		100%

Course Organization

1. There will be several quizzes and several assignments usually about one a week. In addition there will be a midterm and a final exam.
2. The work you turn in on quizzes and exams must be entirely your own work (all quizzes and exams will be closed book and closed notes). However, you are allowed to discuss the programming assignments with any other member of the class or with any PL/C consultant. The final product on each programming assignment must be substantially your own work.
3. We hope to replace each Friday's lecture with two or three section meetings, each of 30-45 students.

Keep every assignment that is handed back to you until the end of the course. Keep a program file for every program you write until the end of the course. (You may be asked to make changes in an earlier program and to resubmit it.)

Keep every quiz and exam that is returned to you. Exam questions will sometimes be based on earlier exam questions or assignments, on the assumption that if you missed something the first time, you should know it by the second.

Programming assignments will be graded by the following criteria:

1. Correctness - Does your program produce the correct answers according to the problem specifications? (Remember, the first step in solving a problem is to make sure that you understand exactly what the problem requires.)
2. Quality - Is your program well-designed and well-structured, and did you design it using the design principles taught in the course? Is it direct and to the point, without useless statements or wasted effort? Is it easy to understand, easy to modify, and easily seen to be correct?
3. Documentation - Are the comments adequate and well-chosen? Have the variable names been chosen to suggest the correct "meaning" of the variables? Is the program listing indented to indicate structure, as taught in class? Does the output look nice and is it labelled well?
4. Generality - How general is your program? (A program that can find the minimum of any number of numbers is better than a program that can only find the minimum of 13 positive numbers.)

Where to use the computer: There are three student laboratories on campus, located in 317 Hammond, 103-104 Boucke, and in the Computation Center.

Lateness Policy: Homework must be handed in on the day it is due. If you have not completed a program by the due date, then hand in whatever you have done with a description of what remains to be done to complete the job. If you have a serious excuse for lateness, speak to the instructor; but remember that your work will not be accepted late, unless you have received permission to hand it in late. Permission to receive a deferred grade will only be granted for strong and compelling reasons.

Grading of Assignments

Introduction to Algorithmic Processes

Each assignment will be graded on a 3 point basis, as follows:

- 3 - good; any deficiencies are minor
- 2 - satisfactory; however, one or more serious deficiencies prevented your work from being graded "good"
- 1 - unsatisfactory; work so seriously deficient that it does not come close to satisfying the assignment
- 0 - zero; work not submitted, or work which does not seem to be a serious effort to do the assignment.

Keep every assignment that is handed back to you until the end of the course. Keep a program deck for every program you write until the end of the course. (You may be asked to make changes in an earlier program and to resubmit it.)

Keep every quiz and exam that is returned to you. Exam questions will sometimes be based on earlier exam questions or assignments, on the assumption that if you missed something the first time, you should know it by the second.

Programming assignments will be graded by the following criteria:

1. Correctness - Does your program produce the correct answers according to the problem specifications? (Remember, the first step in solving a problem is to make sure that you understand exactly what the problem requires.)
2. Quality - Is your program well-designed and well-structured, and did you design it using the design principles taught in the course? Is it direct and to the point, without useless statements or wasted effort? Is it easy to understand, easy to modify, and easily seen to be correct?
3. Documentation - Are the comments adequate and well-chosen? Have the variable names been chosen to suggest the correct "meaning" of the variables? Is the program listing indented to indicate structure, as taught in class? Does the output look nice and is it labelled well?
4. Generality - How general is your program? (A program that can find the minimum of any number of numbers is better than a program that can only find the minimum of 13 positive numbers.)

Where to use the computer: There are three student laboratories on campus, located in 317 Hammond, 103-104 Boucke, and in the Computation Center.

Lateness Policy: Homework must be handed in on [Ⓢ] the day it is due. If you have ← not completed a program by the due date, then hand in whatever you have done with a description of what remains to be done to complete the job. If you have a serious excuse for lateness, speak to the instructor; but remember that your work will not be accepted late, unless you have received permission to hand it in late. Permission to receive a deferred grade will only be granted for strong and compelling reasons.