Scientific Computing – ACMS20210 – CRN 15146

Course Summary ——

An introduction to solving mathematical problems using computer programming in high-level languages such as C.

Prerequisites – Math 10560 -or- Math 10860 -or- Math 10360
Book – Engineering Problem Solving, third edition by Etter and Ingber.
Time and place – MWF 2:00-2:50, Pasquerilla 107.
Instructor – Daniel Brake. Office 152B Hurley. `dbrake@nd.edu` Office hours – Wednesday 9:30-10:30, Friday 10:30-11:30, and by appointment.

Tutorial ——

Assistant – Dawn Lu. Office 215 Hayes-Healy. `Dong.Lv.2@nd.edu`
Dawn's office hours – Mondays, 9:30-11:30
Meeting times – Fridays, 9:25-10:15, 12:50-1:40

Grade ———

Homework	100
Midterm 1	100
Midterm 2	100
Final exam	100
Project	100
Total	500

Homework – assigned weekly, and turned in by Sakai, generally on Monday afternoons. The number of points on homework assignments will not sum to 100. Rather, you will be given a number of points toward your final score, which are the percentage you earned out of the total number available. See the Policies section for information about group collaboration.

Exams – two midterms and a final. Dates for the midterms will be announced in class and online at least two weeks prior. The final is Wednesday, December 16 from 7:30-9:30 pm. A student who misses an examination will receive zero points for that exam unless he or she has written permission from the Vice president for residential life. In such a situation, please contact instructor before the exam to schedule a makeup exam.

Project – a term project, which will allow you to explore some facet of computing which interests you. It will consist of a topic decision, a rough draft writeup, a final code deliverable, a final project writeup, and a short presentation.

Letter grade – will be assigned on the percentage of points you earned, out of the total available, with a scale no more stringent than the typical 90-80-70 scale. The cutoffs for $\pm$ grades are $\pm 3\%$ of the respective letter grade boundary.

Policies ——

Group work – Group work to solve homework problems is acceptable. However, **you must write your own programs and type your own solutions**. Copypasta submission of group work is NOT acceptable, and is a violation of the ND Honor Code, even if you helped write the code in the first place. **You must type your own code**.

ATTENDANCE – You are expected to attend every lecture and tutorial session in which you are enrolled. Lack of attendance is not a valid excuse for missed material or course announcements. If you must miss a lecture for a valid reason, it is your responsibility to obtain notes from a reliable classmate. Attendance will be taken.

HONOR CODE – The ND Honor Code is in full effect for all work related to this course. The instructor takes this seriously. You may not submit code from online or any other source as your own, including group work. Any code you submit must have been typed in by you, personally, physically, using your own fingers on a keyboard. Submission of code developed as a group without typing it yourself, is a violation of the honor code. You must type your own code. Your must type your own code. **You must type your own code**.

ON LEARNING C++ – The C++ language is *very* hard to master, and just as difficult to get started with. You will be challenged. Simple tasks can be confusing. Compiler errors can be opaque, and runtime bugs mysterious. To overcome this barrier, you must personally engage in the struggle that is programming. Class sessions cannot make you a good programmer any more than a band teacher can make you a good violinist. Practice and struggle with instruments and tools are what make a person great, not being a passive observer of instruction. Do not expect to be able to sit down and solve homework problems quickly, even simple ones, merely because you came to class, read the class notes, or saw someone else solve the problem. Classes and slides will be insufficient for completing homework; you will have to get better at independent research, in the book and online. Furthermore, only by starting homeworks soon after their assignment, and allowing yourself plenty of time to complete them, can you hope to succeed in this class. While the instructor is sympathetic to the challenges associated with learning a compiled programming language, especially as a first exposure to programming, the instructor is not sympathetic to low or late effort. Be active in your learning – ask questions, will to succeed despite repeated failure, and be perseverant.

LATE WORK – Late homework is not accepted, except for pre-arranged university excused absences.

Detailed Course Syllabus —————————————————————————————
Broadly speaking, we will cover the entire textbook.

- Introduction to Computing and Engineering Problem Solving – Chapter 1
- Simple C++ Programs – Chapter 2
- Control Structures: Selection – Chapter 3
- Control Structures: Repetition – Chapter 4
- Working with Data Files – Chapter 5
- Modular Programming with Functions – Chapter 6
- One-Dimensional Arrays – Chapter 7
- Two-Dimensional Arrays – Chapter 8
- Intro to Pointers – Chapter 9
- Advanced Topics – Chapter 10

If there are topics in the book which you would like the instructor to cover in particular depth, just ask.