CSF111: Computer Programming

Course Handout (Part-II) — 2023

In addition to Part I (General Handout for all courses appended to the timetable), this portion gives further specific details regarding the course.

Course information

- IC: Swaroop Joshi
- Co-instructors: Anup Basil Mathew and Arnab Kumar Paul
- Lab instructors: Swati Agarwal Hemant Rathore Rizwan Parveen

Introduction

General topics of this course are as follows:

Basic Model of a Computer; Problem Solving-Basic Computing Steps and Flow Charting (Assignment, Sequencing, Conditionals, Iteration). Programming Constructs – Expressions, Statements, Conditionals, Iterators/Loops, Functions/Procedures; Data Types – Primitive Types, Tuples, Choices (Unions or Enumerations), Lists/Arrays, Pointers, and Dynamically Allocated Data. Input-output and Files.

Laboratory Component: Programming Exercises involving development and testing of iterative and procedural programs using bounded and unbounded iterations, function composition, random access lists, sequential access lists, dynamically allocated lists, and file access.

Learning outcomes

After completing the course, students will be able to

- Use basic coding features provided by high-level imperative programming languages
- Write computer programs given simple algorithms
- Analyze simple real-life problems and choose appropriate algorithms to solve them algorithmically
- Use simple data structures like arrays and lists in simple programs
- Read and program using simple APIs provided
- Reason about simple programs
- Develop a viable notional machine for a high-level imperative programming language

- Test simple programs the students have implemented
- Use an industry-strength development environment (Visual Studio Code)

Prerequisites

- None

Electronic resources

- LMS: quanta will be used for submitting labs and displaying marks and some announcements.
- Slides: will be made available via Google Drive.
- Slack: We will use slack as the primary communication channel. We will send messages to everyone in the class, such as corrections to assignments, changes to due dates, clarifications, etc., through slack. Students are required to check slack regularly.
- Programming tools: We recommend using the latest version of Visual Studio Code, gcc, Python3, and the VSC C/C++ plugin.
- Email: your questions are more likely to be answered on a slack channel than a personal email. However, please include “CSF111” in the subject line if you must email me.

Office hours

- Swaroop: WF 11:00-11:45 in D-161 or by appointment (https://calendly.com/swaroopj)
    - Use the calendly link to schedule a 15-minute appointment. This tool has greatly reduced the back and forth for finding a meeting time. If you find a vacant spot on my schedule in that tool, you can make an appointment, and you don’t have to wait for any confirmation. (If I have to change such an appointment for some reason, I will email you as soon as possible.) A plain email asking to schedule an appointment will most likely be ignored.

Text

- J.R. Hanly and E.B. Koffman, “Problem-Solving and Program Design in C”, Pearson Education, Fifth Edition 2007.

Reference books

- R.G. Dromey, “How to Solve it by Computer”, Pearson, 2006.

Evaluation

A **tentative** distribution of evaluative components:

- 10% In-class activities and quizzes
- 25% Programming Labs
- 65% Exams (25 midterms + 40 finals)

Late submission and make-up policy

- In-class activities and quizzes: No late submission is allowed. The lowest *k* items will be dropped to accommodate missed classes. No make-up is allowed.
- Labs: No late submission is allowed. The lowest *k* items will be dropped to accommodate missed labs. No make-up is allowed.
- Exams: Conducted as per the institute timetable. No make-up is allowed without the prior written approval of the IC. IC has the discretion to deny a make-up request.

Attendance

By enrolling in this course. You are implying your availability to complete and attend all lectures, labs, and meetings. There will be no extensions on assignments due to absence unless there is a legitimate documented emergency. Specifically, vacation time and job responsibilities are not legitimate emergencies. You are responsible for all class lectures, including handouts and notes.

Academic misconduct

The fundamental principle determining the scope of acceptable collaboration is that it is never permissible to pass off as your own the work of someone else. If you have doubts about what is appropriate, ask your instructor for a ruling in advance. Violations are surprisingly easy to detect and are dealt with according to the institute's rules on academic misconduct. All students are expected to complete the work while observing the institute's standards for academic integrity. The term "academic misconduct" includes all forms of student academic misconduct wherever committed, illustrated by, but not limited to, cases of plagiarism and dishonest practices in connection with examinations.

A rule of thumb: you should be able to fully explain every part of the answer (including but not limited to a piece of text or code) you have provided for any evaluative component. Failing this, it will be considered academic misconduct, and the student will be reported.

Any student or a group of students suspected of academic misconduct will be reported to the Disciplinary Committee with the recommendation of an NC grade.

Special accommodations

If you need special accommodations for any reason (including but not limited to disabilities, religious beliefs, family situations, etc.), you are welcome to reach out to me so we can go over your situation and plan for the accommodations. But please contact me as early as possible, so we have enough time to make such plans. Accommodations will be granted at the discretion of the IC.

Course plan

The schedule below is **tentative** and is subject to change. However, we will cover these topics in some form or the other.

Lectures	**Topics**	**References***
1	Introduction	Notes,
2-5	Expressions, Operators	Notes; T 2
6-8	Functions	Notes; T 3
9-12	Notional machine, Design Recipe, Testing	Notes
13-15	Conditionals	Notes; T 4
16-20	Strings, Lists, Loops	Notes; T 5, 9
21-24	User-defined data	Notes; T 11
25-28	Recursion	Notes; T 10
29-32	Pointers	Notes; T 14
33-35	File IO	Notes; T 12
36-38	Arrays	Notes; T 8
39-40	How the library was implemented	Notes

* Notes = class notes. T = textbook; numbers denote chapters in the textbook.

Cmd line