CS 101
Computer Programming and Utilization

Instructor: Rohit Gurjar

All material based on Prof. Abhiram Ranade's book and slides
(and previous course instructors)

This lecture

- Introduction to the topic
- Administrative details
- A simple program

Computer/Computing

- Everywhere: smartphones, laptops, TV, video games, aeroplanes, drones, …
- Computing:
  - When you book a train ticket
  - Search on internet
  - Find the shortest route on maps
  - Instagram filter
  - Sending a message on chat
  - Creating a meme
- Goal of the course: How to make computer do such things

What is a computer?

- A computer is a giant electrical circuit that can be instructed/programmed to do various tasks
  - Receive data from the external world
  - data = numbers, images, sounds, can be represented using numbers and fed to a computer, maps, data from sensors,
  - Process the data it receives
  - Send the results back to the external world
- What process/computing is performed?
  - Determined by a program loaded in the computer

Program

- Program: a precise description of the computation we want the computer to perform
- By feeding different programs to a computer you can make it do different computations.
- This course tells you how to construct (“write”) programs.
- Programming language: a language (grammar, vocabulary) to give instructions computers.
- Why not natural languages (English, Marathi)?
  Because ambiguous, not standardised, not expressive enough.
- Exercise: Try describing the procedure of dividing a number by another in English/

Programming for engineers/scientists?

- For example, you will be writing programs
- CSE: to make computers more energy efficient, to verify if softwares/apps developed by others work as desired, to make computers more intelligent.
- EC: to analyze financial data, to build economic models, to simulate economies and forecast
- MM: simulations to predict how materials evolve under certain conditions, predict material properties based on composition

Programming for engineers/scientists?

- For example, you will be writing programs
- CL: to build and simulate mathematical models of reactors, entire plants, to build automated control systems for Chemical plants
- EE: to create and test circuit designs virtually, signal processing
- MA: to solve differential equations, integrals via numerical methods, to visualize mathematical concepts, testing conjectures

How do you get good at programming?

- By doing it yourself
- Almost all learning will happen in the lab.
- TAs will help if you are stuck. You can also discuss ideas with your friends. But, don't directly ask for solutions.
- Write the program yourself.
- Initially, it may be confusing and frustrating. Be patient and continue practicing.
- Prohibited: finding solutions on internet, ChatGPT, Claude, Gemini, Perplexity etc.

Administrative details

Schedule

- Lectures:
  - D2 : LA 201 (slot 11) Tue, Fri 3:30 - 4:55 PM
  - D4 : LA 202 (slot 9) Mon, Thu 3:30 - 4:55 PM
- Weekly lab: (SL1/SL2)
  - P10, P20 : Mon 9:30-11:30 AM
  - P8, P19 : Tue 9:30-11:30 AM
  - P9, P22 : Thu 9:30-11:30 AM
  - P7, P21 : Fri 9:30-11:30 AM
- Extra help session: Sat 2:00-4:00 PM.
- Holidays: Make up lab for 15 Aug, 5 Sep (Fri) on next day (Saturday 2-4 pm)
- Quizzes (written): Aug 20, Oct 15 (Wed) 8:30-9:10 am.
- Lab exams: 8 Sep-12 Sep (respective Lab slots), 3 Nov - 7 Nov (respective lab slots)
- Midsem, Endsem Exams: will be scheduled by TTC.

People

- Instructor: Rohit Gurjar
  Email: rgurjar@cse.iitb.ac.in
- Course Manager: Firuza Aibara (on leave till mid sem)
  Email: firuza@cse.iitb.ac.in
- Managing TA: Sarfaraz Equbal
  Email: sequbal@cse.iitb.ac.in
- 5 RAs (cs101-ras@cse.iitb.ac.in) managing the lab sessions.
- A group of 10-12 students will have a TA.
- Each of the 8 lab sessions will have a Head TAs.
  - Will be shared next week.

Communication

- Bodhi Tree (robin.bodhi.cse.iitb.ac.in): weekly lab submission, lab exams, announcements, marks, discussion forum.
- Course Webpage (https://www.cse.iitb.ac.in/~cs101/): All lecture slides and C++ programs.
- For any doubts from lectures/discussions, use BodhiTree discussion forum, or reach instructor after the class, or email.
- For any issues related to lab, marks in exams etc, Email/talk to head TA.
- For any other issues, Email/talk to instructor and managing TA.

Attendance

- Lectures:
  - As per institute rules, minimum 80% attendance required.
  - If attendance is below 80%, you get DX grade and you have repeat the course.
  - A TA will take attendance in every class from second week onwards.
- Weekly lab:
  - Each lab (from second week onwards) will have some marks which will count towards your grade.
  - Marks are for attending and making a genuine attempt at the problem, getting the correct solution or something close it.
  - If absent for medical or other emergency, email to Head TA with documentation

Evaluation

- Quizzes : 5 + 5 %
- Lab exams: 15 + 15 %
- Midsem: 20 %
- Endsem: 30 %
- Weekly labs: 10 %
- For any quizzes/exams missed due to medical reasons, there will be a make-up exam (theory+lab mix) after the endsem.

Academic Honesty

- Do not copy code from any source (classmates, Internet, ChatGPT, ...); do not send code to each other.
- Every line of submitted code must be written by yourself.
- You can consult references for syntax, formats, etc.
- You can discuss code, concepts with friends.
- When in doubt about taking some action, ask TAs or instructor.
- Do not copy in the tests and exams.
- Suspected academic malpractice will be reported to D-ADAC.

Resources

- Textbook: “An introduction to programming through C++”, Abhiram Ranade, McGraw Hill Education, 2014.
  - Integrated with use of simplecpp
  - will be shared on BodhiTree
- Abhiram Ranade’s NPTEL course
  - https://www.youtube.com/playlist?list=PLEAYkSg4uSQ2qzihjdDEseWrrY1DyxH9P

Give us Feedback

- Initial survey (on CS101 homepage)
  - Please let us know if you need a scribe or any help with other issues.
  - Language issues.
- Link for continuous anonymous feedback (on CS101 homepage)
- TALK to me, TAs.
- We want you to enjoy this course, learn from it, apply your learning to future courses and projects.