

CS 240: Programming in C

Lecture 1: Course Introduction

Lecture 1

- Objectives
- Course policies

Announcements

- Read Chapter 1 in K&R

Lab This Week

- We do have lab this week!
- A re-introduction to the UNIX command line
 - and GCC
- Only time the assignment will be submitted and due during lab!

Instructor

Dr. Christopher May

may5@purdue.edu

Office: DSAI 1119E

Office Hours: MW 3:00 - 4:00 (or by appointment)

Teaching Assistants

- Complete list with office hours on website
- Don't expect TAs to do your homework for you
- Ask TAs about your code -- don't ask them to come up with code for you

Course Website

- <https://mandalore.cs.purdue.edu/~cs240>
- Go here for
 - Syllabus
 - Lecture videos and slides
 - Slides will be available before lecture
 - Videos will be posted one week after lecture
 - Homeworks
 - Schedule

Syllabus

- You are responsible for reading and understanding the entire syllabus
- Don't hesitate to ask questions!

Things You Should Already Know How To Do...

- How to write and compile a Java program in the UNIX environment
 - Using javac, java
- How to declare variables
- How to manipulate arrays
- How to write loops
 - for, while, do-while
- How to use if statements
- How to use switch statements

Things You Should Already Know How To Do...

- Logical operators `|` `|` `&&` `!`
- How to write data with `printf()`
- How to write functions

- How to copy, move, execute, and delete files
- How to navigate a file system (`pwd`, `cd`, etc.)
- How to edit files in a UNIX environment
 - `vim`, `emacs`, `nano`, etc.

IDE

- We will not use an IDE for this class
 - (e.g., Visual Studio, CLion, etc.)
- You are expected to develop on Purdue systems, such as `data.cs.purdue.edu` or `borgNN.cs.purdue.edu`
 - Can do so remotely with Secure Shell (SSH)

Questions and Contact

- Ed Discussion
 - <https://edstem.org/us/join/Bp4CrK>
 - Announcements will be posted here
- Lab
- Office Hours
 - All office hours are in DSAI B069
 - Schedule: <https://mandalore.cs.purdue.edu/~cs240/officehours.php>
- Email
 - <https://mandalore.cs.purdue.edu/~cs240/contact.php>

Lab

- Environment to work on your homework with help available
- The first hour of each lab is reserved for students in that section only
 - Students of that section always have priority
 - TAs reserve the right to ask people to leave
- Attendance is optional, after the first week

Homework Assignments

- Usually due on Wednesdays at 9PM
- Usually assigned on Monday of the previous week
 - The overlapping 3 days are there as a buffer to help you manage your time
 - The expectation is that you are done with assignments no later than the weekend
- Approximately 12 assignments, 100 points each
- Style grade worth additional 20 points

Code Standard

- Intended to serve as basic recommendations and requirements for how you write your code
- Like a dialect or accent of a language
- All good software development environments use a code standard
- You may not agree with it, but you are graded on it
- <https://mandalore.cs.purdue.edu/~cs240/code.php>

Test Modules

- You will be provided with a test module for most assignments
 - Can run it as many times as you want
- If your program crashes, you will receive a score of 0
- Test module will not progress to the next function until all tests up to that point pass
- Run your test modules often
 - Many tests are randomized
 - We will run it many times for grading and take the lowest score

Submission

- Programs that do not compile will not receive credit
- We will always grade the latest submission
- Only run “make submit” if you want us to grade the current version using the current date and time

Regrades

- Aside from exceptional circumstances, regrades are typically honored only when a bug in the test module is discovered
 - The entire course is then regraded
- Always compile and test your programs before submitting
 - Even for one line changes

Help

- Be prepared to talk about steps you have already taken in debugging
- Avoid questions like “this doesn’t work. Why?”
- Instead, “I am having trouble with part X. I have printed out the values of variables Y and Z on lines K and N. I set a breakpoint for function F and stepped through to line W. I think that the issue involves variable Z...”

Ed Discussion

- Great resource for posting questions about homework and other course content
- Any post involving code **must** be private
- We expect people to treat each other with respect
- Work to keep a positive atmosphere

!Ed Discussion

- Ed Discussion is NOT a place for
 - Regrade requests
 - Submit through Gradescope for quizzes and code standard
 - Complaints
- Posts that are not productive or overly negative will be removed

Quizzes

- After-lecture quizzes
 - 25 points, submit no later than 24 hours after lecture
- In-lecture quizzes
 - 10 points, submit during lecture
- No submission? Score of 0 unless excused
- See syllabus for more information

Grades

- Homework and Quizzes: 50%
- Two midterms: 14% each
- Final: 22%

Grading issues will be addressed as they occur. Not at the end of the semester.

Grade Determination

| Homework/Quiz Avg. | | Exam Avg. | | Course Avg. | Grade |
|--------------------|------------|-------------|------------|-------------|----------------|
| $\geq 85\%$ | and | $\geq 85\%$ | and | $\geq 90\%$ | A |
| $\geq 75\%$ | and | $\geq 75\%$ | and | $\geq 80\%$ | B |
| $\geq 65\%$ | and | $\geq 65\%$ | and | $\geq 70\%$ | C [*] |
| $\geq 55\%$ | and | $\geq 55\%$ | and | $\geq 60\%$ | D |
| $< 55\%$ | or | $< 55\%$ | or | $< 60\%$ | F |

*C threshold is often lowered

Academic Integrity

- If you claim someone else's work as your own, it's considered cheating
- Do not do quizzes or exams for other students
- Asking questions is probably OK, but do not look at other students' homework files and do not work with other students
- Protect your work
 - If your work is turned in by another student, you are both guilty

Academic Integrity

- When in doubt, talk to a TA or instructor instead of a fellow student
- All work is subject to computer-based comparison and analysis

AI/ML/LLMs/etc.

- Using ChatGPT and similar software to generate partial or complete solutions to assignments, quizzes, exams, etc. is also cheating
- Think of it like using a calculator instead of learning basic arithmetic
 - You're only hurting yourself and future career prospects

Academic Integrity

- The minimum penalty for any incident of academic dishonesty is a score of zero for the item in question
- More serious or repeated infractions will result in a grade of 'F' for the course
- All incidents will be referred to the Dean of Students and the Department of Computer Science

Academic Integrity

- Ask questions about the policy *without risk*

Supplemental Instruction

- <https://mandalore.cs.purdue.edu/~cs240/supplement.php>
- Data shows students that regularly attend SI sessions obtain higher grades in the course, on average

Tips for Success

- Start on homework projects as early as possible
- Read the textbook
- Practice
- Experiment
- Create a cool personal project
- Look at source code available on the internet
- If you can learn to enjoy programming you are guaranteed to do well!

How to Fail This Course

- Assume that since your prerequisites were easy, this class will be too!
- Try to do the homework at the last minute
- Don't do the homework at all
- Don't come to lecture
- Get “help” from somebody else in the class
- Don't practice for the exams

Why C?

- It is still widely used *and growing*
 - [TIOBE](#) August 2024 index lists it as #3
 - Python is #1, C++ is #2
- Programming Language of the Year 2019
- Used in a huge number of embedded and IoT devices
- Ubiquitous for systems programming
- Small
- Fast

C vs. Java

- You already know how to write some C code
 - Java was designed using C/C++ style syntax
- But, there are *many* differences

C vs. Java

| Attribute | C | Java |
|--------------------------------|---------------------------------|----------------------------|
| language type | function oriented | object oriented |
| basic programming unit | function | class (ADT) |
| source portability | yes, when done right | yes |
| compiled binary portability | recompile for each architecture | “write once, run anywhere” |
| security | yes, when done right | more built-in safety |
| compiling | gcc hello.c | javac Hello.java |
| linking a library (e.g., math) | gcc -lm mycalc.c | javac MyCalc.java |
| execution | a.out (or name of binary) | java MainClass |

C vs. Java

| Attribute | C | Java |
|------------------------------|---|---|
| (runtime) array declaration | <code>int* a = malloc(N * sizeof(a));</code> | <code>int[] a = new int[N];</code> |
| array size | often unknown | <code>a.length</code> |
| strings | '\0'-terminated char array | built-in immutable String type |
| using a library | <code>#include <stdio.h></code> | <code>import java.io.File;</code> |
| accessing a library function | <code>#include <math.h></code> <code>x = sqrt(2.2);</code> | <code>import java.util.Math;</code> <code>x = Math.sqrt(2.2);</code> |
| standard output | <code>printf("sum = %d\n", x);</code> | <code>System.out.printf("sum = %d\n", x);</code> |
| reading from stdin | <code>scanf("%d", &x);</code> | <code>int x = StdIn.readInt();</code> |

C vs. Java

| Attribute | C | Java |
|-----------------------|-----------------------|------------------------------|
| memory address | pointer | reference |
| manipulating pointers | *, &, + | not permitted |
| functions | int max(int a, int b) | public int max(int a, int b) |
| data structures | struct | class |
| methods | function pointers | yes |
| pass-by-value | yes | yes |
| allocating memory | malloc() | new |
| de-allocating memory | free() | garbage collection |

C vs. Java

| Attribute | C | Java |
|------------------------------|-------------------|----------------------------|
| constants | const and #define | final |
| variable auto-initialization | not guaranteed | yes, compile-time checking |
| data type for generic item | void* | Object |
| variable naming convention | sum_of_squares | sumOfSquares |
| file naming convention | stack.c, stack.h | Stack.java |
| assertions | assert | assert |

Not a complete list!

For Next Lecture

- Read and review Chapter 1 of your textbook
- Familiarize yourself with C programming
- Ensure your computer account is setup
- Choose an editor, practice with it
- Practice writing a few programs
 - “Hello, world”
 - Prompt for a string, print it back
 - Write something to do your calculus homework

Slides

- Slides are heavily based on Prof. Turkstra's, Prof. Rodriguez-Rivera's, Prof. Fred Mowle's, and Dr. Richard Kennell's material