Lecture P1: Introduction to C



#include <stdio.h>
int main(void) {
 printf("This is a C program.\n");
 return 0;

C Background

Born along with Unix in the early 1970s.

- One of most popular languages today.
- Basis of C++ and Java.

C Features.

- . Concise.
- Widespread usage.
- . Exposes low-level details of machine.

Consequences.

- Positive: you can do whatever you want.
- Negative: you can do whatever you want.

0



Programming is learned with practice and patience.

- Don't expect to learn solely from these lectures.
- . Do exercises.
- . Experiment and write lots of code.

Do reading.

- Read King Chapters 1-6 today!
- Read again over the weekend.

Aspects of learning to program.

- Language syntax.
- . Algorithms.
- Libraries.
- . These are different skills and learning processes.



Anatomy of a While Loop

Print a table of values of function $f(x) = 2 - x^3$.

• Use while loop to perform repetitive tasks.



Language Syntax: Loops

Print a table of values of function $f(x) = 2 - x^3$.

• Use while loop to perform repetitive tasks.









Library Functions: printf()

Library functions.

. Functions provided as part of C implementation.

Example: printf().

- . Contact between your C program and outside world.
- Puts characters on "standard output."
- By default, stdout is the screen that you're looking at.

Internally, all numbers represented in BINARY (0's and 1's).

. printf() displays more useful representation (int, double).

Formatted output.

- . How do you want the numbers to look?
 - integers, how many digits?
 - real numbers, how many digits after decimal place?
- . Very flexible.



Library Functions: rand()

Print 10 "random" integers.

 Library function rand() in stdlib.h returns integer between 0 and RAND_MAX (32,767 = 2¹⁶ - 1 on arizona).



Library Functions: rand()

Print 10 "random" integers between 0 and 599.

- No precise match in library.
- . Try to leverage what's there to accomplish what you want.





20





Gambler's Ruin Simulate gambler placing \$1 even bets. Q. How long does the game last if we start with \$c ? Unix To print plot, replace: % gcc gambler.c % a.out printf("%d\n", cash); % a.out 4 1234 4 543 *** *** with **** ** **** *** **** **** i = cash;*** *** **** **** while (i > 0) { *** printf("*"); ** i--; printf("\n"); ******* *******

24

Gambler's Ruin Numerical Experiment

Goal: run experiment to see how long it takes to go broke.

Do for different values of starting cash values c.

	Unix								
	% gcc gexperiment.c % a.out			# bet	# bets				
	2	2	6	304	2	2			
initial cash	3	33	17	15	53	29			
	4	22	1024	7820	22	54			
	5	243	25	41	7	249			
	6	494	14	124	152	14			
	7	299	33	531	49	93			
	8	218	10650	36	42048	248			
	9	174090315	83579	299	759	69			

25



Gambler's Ruin Numerical Experiment





Gambler's Ruin Numerical Experiment

	Unix								
	% <u>c</u> % a	gcc gexperi a.out							
	2	2	6	304	2	2			
initial cash	3	33	17	15	53	29			
	4	22	1024	7820	22	54			
	5	243	25	41	7	249			
	6	494	14	124	152	14			
	7	299	33	531	49	93			
	8	218	10650	36	42048	248			
	9	174090315	83579	299	759	69			

How long will it take to go broke?



Layers of abstraction.

• Random bit \rightarrow gambler's ruin sequence \rightarrow experiment.

Programming Style

Concise programs are the norm in C.

Your goal: write READABLE and EFFICIENT programs.

- . Use consistent indenting.
- Choose descriptive variable names.
- . Use comments as needed.

"Pick a style that suits you, then use it consistently."

-Kernighan and Ritchie

Programming Advice

Understand your program.

- . What would the machine do?
- . Explain it to the teddy bear.

Read, understand, and borrow from similar code.



" Good artists borrow. Great artists steal."

Develop programs incrementally.

- . Test each piece separately before continuing.
- Plan multiple lab sessions.

Summary

Lots of material.

C is a structured programming language.

- . Functions, loops.
- Simple, but powerful tools.

Programming maturity comes with practice.

- . Everything seems simpler in lecture and textbooks.
- . Always more difficult when you do it yourself!
- . Learn main ideas from lecture, learn to program by writing code.

You will create many bugs without any practice whatever.

"As soon as we started programming, we found out to our surprise that it wasn't as easy to get programs right as we had thought. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs."

--Maurice Wilkes, 1949

33

Lecture P1: Extra Notes









What is a C Program?

C PROGRAM: a sequence of FUNCTIONS that manipulate data.

. main() function executed first.

A **FUNCTION** consists of a sequence of DECLARATIONS followed by a sequence of STATEMENTS.

- Can be built-in like printf(...).
- . Or user-defined like f(x) or sum(x, y).

A DECLARATION names variables and defines type.

- . double double x;
- . integer int i;

A STATEMENT manipulate data or controls execution.

- **assignment:** x = 0.0;
- control: while (x < 2.0) {...}
- function call: printf(...);