



Oregon State  
University

COLLEGE OF ENGINEERING

| School of Electrical Engineering  
and Computer Science

## CS 161

### Introduction to CS I

- Ready for Assignment 1
- How do computers interact with users?
- How do we calculate new information?



# Tips

- Lecture slides are posted after each lecture on the website
  - Be sure you have the correct website!
  - <http://classes.engr.oregonstate.edu/eecs/winter2020/cs161-020>
- Sign up for demos using your <onid>@oregonstate.edu address
  - ONID: login (e.g., wagstafk) not id number
- Use Piazza for questions, but **don't post code there**
- Revision Plan: Extra credit here cannot cause you to exceed the total points available
  - (But within the assignment it can)
- Class structure: more reading in early weeks, less later (more hands-on work)
- Demos for Assignments 1-4 (not 5)
  - Do not sign up in week 10 (you will be busy with other things anyway!)

# Positive community

<http://classes.engr.oregonstate.edu/eecs/winter2020/cs161-020/links/community.html>

## CS 161 - Introduction to Computer Science I

Winter 2020: MWF 2 - 2:50 p.m., LINC 228

Home

Syllabus

Calendar

Assignments

Labs

Useful Links

Student Clubs

TA Bios

### Establishing a positive community

Every student should feel safe and welcome to contribute in this course. As the instructor, I will try to establish this tone whenever possible, but ultimately the responsibility for cultivating a safe and welcoming community belongs to the students—that means you!

Fortunately, being part of a safe and welcoming community is not hard. A good place to start is to recognize (and continually remind yourself) of the following facts:

1. Your classmates come from a variety of cultural, economic, and educational backgrounds. Something that is obvious to you may not be obvious to them.
2. Your classmates are human beings with intelligence and emotions. This applies even when sending emails or posting messages on Piazza.
3. Your classmates are here to learn. They have the right to pursue their education without being distracted by others' disruptive behavior, or made uncomfortable by inappropriate jokes or unwanted sexual interest.

# Syllabus quiz

- Open notes!
- Open syllabus!

You will ace this!

## Syllabus quiz

1. What device do you need to bring to lab?
2. What is one thing you can do to create a positive learning environment?
3. What is the demo period for Assignment 1?  
(from \_\_\_\_\_ to \_\_\_\_\_)
4. If you send email, what should you put at the start of the subject line?
5. When can you work together on an assignment?



# C++ primitive types

- Whole numbers: `short`, `int`, `long`: 27, -96323423, 0
  - Can also be “`unsigned`”
- Real numbers: `float`, `double`: 3.14159, -27.0, 2.4e5
  - `float` range: 1.2e-38 to 3.4e38
  - `double` range: 2.2e-308 to 1.8e308
- Characters: `char`: 'H', '2', '%', 'r'
- Boolean: `bool`: true, false
- Later you will learn how to create your own data types

# Smallest and largest numbers

## Unsigned (Positive)

Bits	# Values	Smallest	Largest
1	2	0	1
2	4	0	3
8	256	0	255
16	65536	0	65535
$b$	$2^b$	0	$2^b - 1$

unsigned short

1/8/2020

## Signed (Half Negative, Half Not)

Bits	Smallest	Largest
1	N/A	N/A
2	-2	+1
8	-128	+127
16	-32768	+32767
$b$	$-2^{b-1}$	$2^{b-1} - 1$

short

long (32 bits) ?

CS 161

7

# Minimum and maximum values

Type	Minimum	Maximum
short	-32,768	+32,768
unsigned short	0	65,535
int	-2,147,483,648	+2,147,483,647
unsigned int	0	4,294,967,295
long	-9,223,372,036,854,775,808	+9,223,372,036,854,775,807
float	1.2e-38	3.4e38
double	2.2e-308	1.8e308

## Let's get started on Assignment 1 – Fortune Teller

- Query the user for 5 numbers
- Use them to fill in (and print out) their fortune
- Decide what data type to use for each number
  - Explain (in comments) why you chose that type
  - State the min/max values of that type
- Follow the style guide:  
<http://classes.engr.oregonstate.edu/eecs/winter2020/cs161-020/assignments/cs161-style-guidelines.pdf>
  - Some items will not be relevant yet. Revisit the style guide for each assignment.



# A possible layout: 2 terminal windows

Editor

Terminal

The image shows two terminal windows side-by-side. The left window is an Emacs editor displaying C++ code for a fortune-telling program. The right window is a terminal window showing the compilation of the code into an executable named 'le3-assign1'. A red arrow points from the text 'Editor' to the Emacs window, and another red arrow points from the text 'Terminal' to the terminal window.

```
File Edit Options Buffers Tools C++ Help
/*****
** Program: 
** Author: 
** Date: 
** Description: 
** Input: 
** Output: 
*****/ 
#include <iostream>

using namespace std;

int main() {

    // Declare variables. For each one, include
    // (1) why you chose the data type for the variable
    // (2) the minimum and maximum value that it can hold

    // Prompt the user to enter a value for each number,
    // and store it in the appropriate variable

    // Print out the user's fortune with blanks
    // filled in by the numbers they entered

    return 0;
}

-UU-:----F1 lec3-assign1.cpp All L2 (C++/1 Abbre
```

```
wkiri@madrone demos % g++ lec3-assign1.cpp -o lec3-assign1.cpp
```

# We will cover new terms and ideas today

- Make a list as we go!

# User input and output

- Read in with **cin**
- Write out with **cout**
  - Which direction do the angle brackets go?
- Good style:
  - Provide a helpful prompt to request input
  - Include a space before the point they start typing
  - Print blank lines to increase readability (use **endl**)

# Constants

- Use the “`const`” keyword when you don’t want to allow changes
- The compiler will enforce this for you
- What is wrong with this program?

```
/*
 * CS 161-020, Lecture 3, Winter 2020
 * Use of 'const' keyword
 * Author: Kiri Wagstaff
 * Date: January 10, 2020
 */
#include <iostream>

using namespace std;

int main()
{
    /* The answer to Life, the Universe, and Everything! */
    const int answer = 42;

    /* I changed my mind */
    answer = 37;

    /* Print the answer */
    cout << answer << endl;

    return 0;
}
```

# Expressions

- Sequence of **operations** that yield a value

**Operand    Operator**

— 1 + 2 + 3

- Evaluate:** convert an expression into its value

—  $(54 / 9) + 17 \Rightarrow ?$

—  $7 \% 2 \Rightarrow ?$

—  $15 \% 12 \Rightarrow ?$

—  $x = 99; \quad x++ + 1 \Rightarrow ?$

## Operators

- Add +
- Subtract -
- Multiply \*
- Divide /
- Remainder %
- Increment ++
- Decrement --

# Operator precedence (Rao Appendix C)

- $3 + 4 * 5 \Rightarrow ?$
- “Binding power” of operator
  - Strongest go first
  - Tie: evaluate left to right
  - Parentheses are your friends!
- Examples
  - $3 * 4 + 5 \Rightarrow ?$
  - $3 * (4 + 5) \Rightarrow ?$
  - $(3 * 4) + 5 \Rightarrow ?$
  - $3 - 2 + 7 \% 3 - 1 \Rightarrow ?$
  - $x = 99 \Rightarrow ?$



Assignment is an operator, too

# Expressions and data types

- Operations on **integers** result in an **integer**
  - $6/3 => ?$
  - $3/2 => ?$
  - (Integer arithmetic)
- Operations on **floats** result in a **float**
  - $3.0/2.0 = ?$
  - (Floating point arithmetic)

# What vocabulary did we learn today?

- Your ideas, plus...
- Constant
- Expressions
  - Operator
  - Operand
  - Evaluate
  - Precedence

# What ideas and skills did we learn today?

- Positive learning environment
- User input and output
- Control variable modification with “const”
- Calculating values with expressions
- Integer vs. floating point arithmetic
- Operator precedence

# Week 1 nearly done!

- Attend lab (laptop required)
- Read **Rao Lesson 5** (pp. 85-91, 104-111) – 14 pages
- Finish **Assignment 1** (due Sunday, Jan. 12)
- Useful reference: <http://www.cplusplus.com/>
  
- More fun: Try out **Edabit**: <https://edabit.com/challenges>
  - CS 161 Week 1 collection: <https://tinyurl.com/cs161-week1>
  - When you finish a challenge, look at other solutions
  - Ensure you select “C++” in the language drop-down (defaults to JavaScript)



# Edabit: Fun programming practice

The screenshot shows the Edabit website interface. At the top, there's a green navigation bar with the Edabit logo, 'Tinker', 'Challenges' (which is underlined), 'Shuffle', and 'Help'. On the right of the bar are 'Sign In' and 'Register' buttons. Below the bar, the page title 'CS 161 Week 1' is displayed. A sidebar on the left lists 'Fun challenges to try in week 1' with a timer icon showing '1'. The main content area contains three challenge cards:

- Return the Sum of Two Numbers**: Create a function that takes two numbers as arguments and return their sum. Examples addition(3, 2) → 5 addition(-3, -6) → -9 addition(7, 3) → 10 Notes Don't forget to return the result. If you get stuck on a challenge, find help in the Resources tab. If you're really stuck, unlock solutions in the Solutions tab.  
Difficulty: Very Easy
- Area of a Triangle**: Write a function that takes the base and height of a triangle and return its area. Examples triArea(3, 2) → 3 triArea(7, 4) → 14 triArea(10, 10) → 50 Notes The area of a triangle is: (base \* height) / 2 Don't forget to return the result. If you get stuck on a challenge, find help in the Resources tab. If you're really st ...  
Difficulty: Very Easy
- Return the Remainder from Two Numbers**: There is a single operator in C++, capable of providing the remainder of a division operation. Two numbers are passed as parameters. The first parameter divided by the second parameter will have a remainder, possibly zero. Return that value. Examples remainder(1, 3) → 1 remainder(3, 4) → 3 remainder(-9, 45) → -9 remainde ...  
Difficulty: Very Easy

# Minute paper

- Please write down on scratch paper:
  1. One thing you learned today
  2. One concept you find confusing
- You can include your name or leave it anonymous
- Leave it in a box at the door as you go out

Thank you!