CS 161
Introduction to CS I

- Prepare you for Assignment 1
- How do we store information in a computer?
- What is good programming style?

Oregon State University
College of Engineering

General tips

- Post your questions/issues/obstacles on Piazza
- If needed, email cs161-w20-ta@engr.orst.edu
- If you need to contact me only, use kiri.wagstaff@oregonstate.edu
  - Do not contact me through Canvas. I may not see it.

Oregon State University
College of Engineering

How to sign up for assignment grading (demo)

http://classes.engr.oregonstate.edu/eecs/winter2020/cs161-020/index.html

TA Information: Office/Grading Hours

All demos are in KEC 1174

Name	Email	Office Hours (DEAR 119)	Grading Hours (KEC 1174)
Sabrina Jesmin	jesmins		
Yipeng (Roger) Song	songyip	By appointments	N/A
Megan Black	blackme	Mon & Wed 12-2 p.m.	Mon & Thu 2-3:50 p.m.
Erick Branner	brannere	Mon 12-2 p.m.	Mon 10 a.m.-12 p.m. & Fri 9-11 a.m.
Jesse Chick	chickj	Tue & Thur 9-10 a.m.	Mon 9-11 a.m. (except Mon 1/20 -> Fri 1/17)
Louis Duvoisin	duvoisil	Mon & Fri 12:00-2:00 p.m. | Tue & Thur 11:30 a.m.-2:00 p.m. | Wed 12:00-1:00 p.m.	N/A
Jessica Garcia	garciaj3	Mon 6-7 p.m. & Wed 8-9 a.m.	Weeks 2-5: Tue 4-5 p.m. & Wed 9-11 a.m.; Weeks 6-10: Tue 2-5 p.m.

How to sign up for assignment grading

- Check timezone and week!
- You can sign up for all 5 demo slots now if you want!
- Important notes
  - Demos outside of 2 weeks receive a 50% penalty for implementation part of the assignment
  - No demo: 0 points for implementation part
  - Take notes during demo if you want to submit a Revision Plan for extra credit on the assignment

Oregon State University
College of Engineering

We will cover new terms and ideas today

- Make a list as we go!

I want to write a program

Oregon State University
College of Engineering

- Hello, humans!

```
#include <iostream>

using namespace std;

int main()
{
  cout << "Hello, humans!" << endl;

  return 0;
}
```

```
  #include <iostream>
 int        main(
) {std::  cout<<
   "Hello, humans!"
 <<std::
endl
    ;     return
 0    ;      }
```

Both work – they are the same to the compiler.
But please use the style on the left (more friendly to humans!)

- Want to write crazy code? https://www.ioccc.org/

Your tools

- Terminal / command line / **shell** – what is it used for?
  - Create, move, delete files
  - Navigate the filesystem
  - Run programs (compiler, editor, your program!)
  - Your shell is called “bash”
- **Text editor** – what is it used for?
  - Create, edit, update programs
  - Your editor is called “vim” (but others are fine too)
  - Syntax highlighting

Oregon State University
College of Engineering

A closer look at our first program

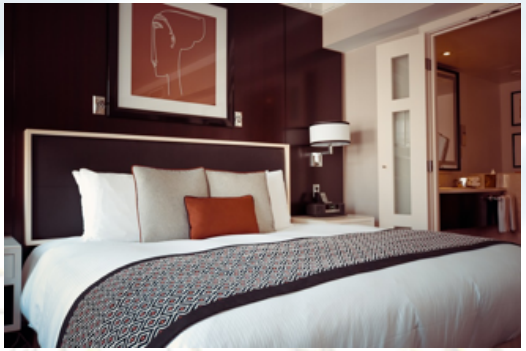
- Tinker/change to see what causes errors
- You cannot fail an experiment!

Storing information in memory

- **Variable**: name for a location in memory whose value can be changed
- **Constant:** name for a location in memory whose value **cannot** be changed
- **Literal**: value, not a variable: `5, "Mars", 3.14159`
- **Identifier:** name for a variable or function
  - Naming rules: see Rao Appendix B for forbidden names

Storing information in memory

- **Declaration** (reserve a hotel room)
- **Initialization** (reserve & check in)
  vs. **assignment** (reserve now, check in later)



The language of computers

- Humans: decimal
- Computers: binary
  - Bit: 0 or 1
  - Byte: 8 bits
  - Count on your fingers
- What about words?
  - Each letter has an 8-bit binary representation (ASCII)

Binary numbers

- How many light switches in your home?
- How many numbers can you encode by turning lights on and off?
  - 💡 : ?
  - 💡💡 : ?
  - 💡💡💡💡💡💡💡💡 : ?
  - b 💡 : ?

Smallest and largest numbers

Bits	# Values
1	2
2	4
8	256
16	65536

Smallest and largest numbers

Bits	# Values	Smallest	Largest
1	2		
2	4		
8	256		
16	65536		
b	2^b		

Smallest and largest numbers

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

Smallest and largest numbers

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

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	$2^b - 1$

Signed (Half Negative, Half Not)

Bits	Smallest	Largest
1		
2		
8		
16		
b		

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`

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)?`

Oregon State University
College of Engineering

C++ primitive types

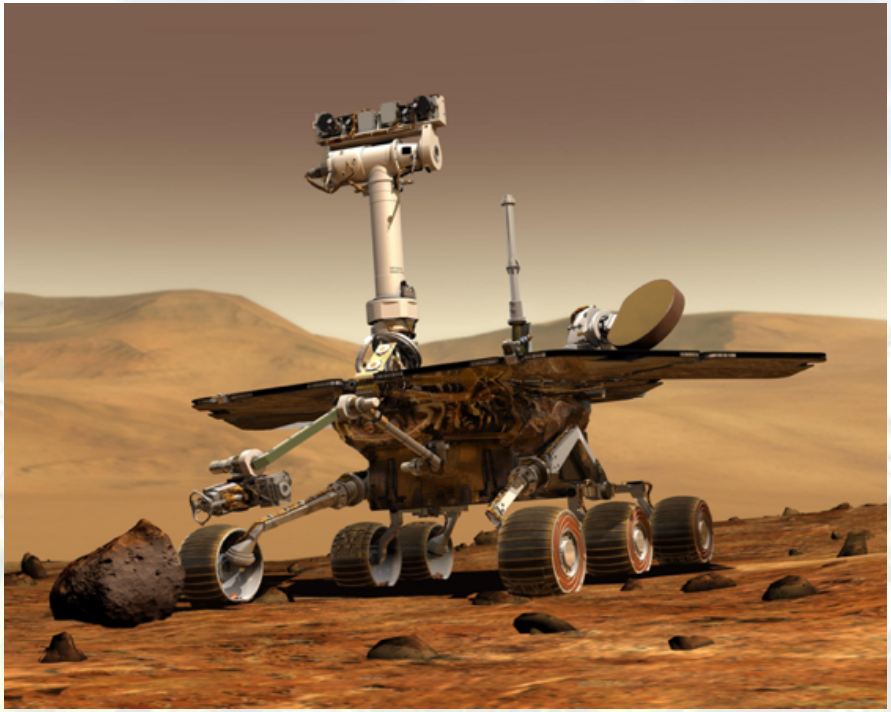
- Whole numbers: `short`, `int`, `long`: 27, -96323423, 0
  - Can also be "`unsigned`"
- Real numbers: `float`, `double`: 3.14159, -27.0, 2.4e5
  - `float` range: 1.2e-38 to 3.4e38
  - `double` range: 2.2e-308 to 1.8e308
- Characters: `char`: `H', `2', `%`, 'r'
- Boolean: `bool`: true, false
- Later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

How to choose what to use?

- Laptop CPU (1.6 GHz), RAM (16 GB)
- iPhone CPU (1.84 GHz, dual-core), RAM (2 GB), storage (32 GB)
  - But who wants to download a bloated app?
- Mars rover CPU (200 MHz), RAM (256 MB)

What type would you use to store…

1. Number of kilometers driven
2. Number of images taken
3. Temperature
4. Sol (day of mission)
5. Age of the Universe



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.

- Questions?

What vocabulary did we learn today?

- Tools
  - Terminal / shell
  - Editor
- Programming
  - Algorithm
  - Constant vs. literal vs. variable
  - Declaration
  - Identifier
  - Standard in and standard out
- Binary numbers
  - Bit
  - Byte
- Data types
  - Primitive
  - Boolean (`bool`)
  - Character (`char`)
  - Integer (`short`, `int`, `long`)
  - Floating point (`float`, `double`)

What ideas and skills did we learn today?

- Decide what C++ data type best fits what you want to store
  - Why does this matter?
- **Declare** variables
- **Initialize** vs. **assign** variables
- Binary numbers: # values, minimum and maximum possible
  - Impact of using "signed"
- Good coding style

On track to finish week 1

- ❑ Read the syllabus – there will be a quiz!
- ❑ Attend lab (laptop required)
- ❑ Read **Rao Lesson 3** (pp. 31-47 + pp. 58-59) -> help for Assign #1
  - ❑ Also review slide 18 of this lecture
- ❑ Finish **Assignment 1** (due Sunday, Jan. 12)
- ❑ Try **Rao Exercise 2.1** (p. 29) – answers at the back of the book

- ❑ 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)

See you Friday! Go forth and conquer!