

# **CS 161**

## **Intro to CS I**

Finish Loops/Begin C++ Strings



## Odds and Ends

- Assignment 2 demoed this week!
- Study sessions Tues, Wed, Thur.

5pm 6pm 7pm — LINC  
268



# How do we read a string of chars?

- User-defined type in string library

```
#include <string>
```

*C++ strings*  
*<string>* *string.h* - *C strings*

- Declare/Create type

```
std::string mssg;
```

*object* ← *container of info + functions*  
*allow us access to those*

- Read with cin or getline

```
cin >> mssg; //get a word
```

```
getline(cin, mssg); //get a line of txt
```

```
2. ENGR
Re-attach Fullscreen Stay on top Duplicate Close
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main() {
7     int i;
8     float f;
9     string s;
10
11     //ignore all leading whitespace, read characters matching my type
12     //until something not of my type or whitespace without moving past
13     //what stopped me from reading
14     cout << "enter a integer: " << endl;
15     cin >> i;
16     cout << "i: " << i << endl;
17
18     cout << "enter a float: " << endl;
19     cin >> f;
20     cout << "f: " << f << endl;
21
22     cout << "enter a string: " << endl;
23     cin >> s;
24     cout << "s: " << s << endl;
25     cout << "length: " << s.length() << endl;
26     cout << "first character: " << s.at(0) << endl;
27
28     return 0;
29 }
```

24,4

All

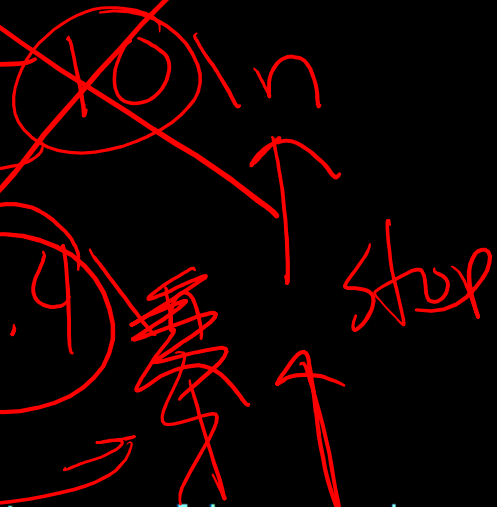
# Reading multiple types of data...



Oregon State University  
College of Engineering

- What does cin do when reading...
  - Int/Floats
  - Strings
  
- What does getline do?

```
3. ENGR
Re-attach Fullscreen Stay on top Duplicate
6 int main() {
7     int i;
8     float f;
9     string s;
10
11     //ignore all leading whitespace, read characters matching my type
12     //until something not of my type or whitespace without moving past
13     //what stopped me from reading
14     cout << "enter a integer: " << endl;
15     cin >> i;
16     cout << "i: " << i << endl;
17
18     cout << "enter a float: " << endl;
19     cin >> f;
20     cout << "f: " << f << endl;
21
22     cout << "enter a string: " << endl;
23     //cin >> s;
24     getline(cin, s); //read characters until I hit a newline, and move
    past
25     cout << "s: " << s << endl;
26     cout << "length: " << s.length() << endl;
26,4 55%
```



```
2. ENGR
Re-attach Fullscreen Stay on top Duplicate
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main() {
7     int i;
8     float f;
9     string s;
10
11     cout << "enter a integer: ";
12     cin >> i;
13     cout << "i: " << i << endl;
14
15     cout << "enter a float: ";
16     cin >> f;
17     cout << "f: " << f << endl;
18
19     cout << "enter your name: ";
20     //cin >> s;
21     getline(cin, s); //move past the newline in the buffer leftover from the float
22     getline(cin, s); //now read a line
23     cout << "s: " << s << endl;
24     cout << "length: " << s.length() << endl;
25     cout << "first letter in name: " << s.at(0) << endl;
26
27
28     return 0;
29 }
-- INSERT -- 22,39 All
```

iversity  
neering



## Extra Credit #2

- Get into groups of 4-5.
- Write your name on one piece of paper.
- How would you determine if user entered a good positive int **without using cin.clear() and cin.ignore()**?

strings

---



```
2. ENGR
Re-attach Fullscreen Stay on top Duplicate
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int i;
7     float f;
8
9     cout << "enter an int: " << endl;
10    cin >> i;
11
12    //This is a way to check the failbit and reprompt the user if cin fails
13    //When will this work and when will it not work?
14    while(cin.fail()) {
15        cin.clear();
16        cin.ignore(256, '\n');
17        cout << "enter an int: " << endl;
18        cin >> i;
19    }
20
21    cout << "enter a float: " << endl;
22    cin >> f;
23
24    return 0;
25 }
-- INSERT -- 13,52 All
```

sity  
ering

Solu

```
2. ENGR
Re-attach Fullscreen Stay on top Duplicate
1 #include <iostream>
2 #include <string>
3 #include <stdlib.h>
4
5 using namespace std;
6
7 int main() {
8     int i;
9     float f;
10    string s;
11    bool error;
12    do {
13        error=false;
14        cout << "enter an integer : " << endl;
15        getline(cin,s);
16
17        for(int i=0; i<s.size(); i++) {
18            if(!(s.at(i) >= '0' && s.at(i) <='9')) {
19                cout << "error!" << endl;
20                error=true;
21            }
22        }
23    }while(error);
24
25    i=atoi(s.c_str());
26    cout << i << endl;
27
28    cout << "enter a float: " << endl;
29    cin >> f;
30
31    return 0;
32 }
2 more lines; before #9 3 seconds ago 29,7 All
```

University  
Engineering

# Programming Errors

- Syntax errors
  - Misuse of C++ language
  - How are they caught?
- Logic errors
  - Doesn't perform task correctly (aka. bugs)
  - How are they caught?
- Runtime errors
  - Stops your program from running
  - How are they caught?

# Syntax Error Examples



- Missing main function
- Use of identifier not declared
- Misspelled Words
- Forget a Semicolon
- Forget Required Keyword
- Missing quote, curly brace, and parenthesis
- Use of single quotes instead of double

# Logic Error Examples



- Poorly written programs
  - Add instead of subtract (incorrect operation)
  - Using last two digits for date
  - Same error message for different errors
  - Program that never ends
  - Add one to the largest integer (could be syntax)

# Runtime Error Examples

- Segmentation fault or Core dump
  - Read a file that doesn't exist
  - Go outside of memory bounds
  - Run out of memory
  - Divide by variable that is zero

# Debugging Errors



- Syntax:
  - **READ compiler errors** (pay attention to line #)
  - Use **google** to search for error
- Logic/Runtime
  - Use **std::cout** to find where the code is breaking
    - **Print variable values**
    - **Print indicator messages**
  - **Trace** through the code
  - **Comment** out code