

CS 161

Intro to CS I

Strings, cin, and errors

Finish C++ String Demo

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

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int x;
6     float f;
7
8     //Not a good way to handle errors because 4.6, 4t, etc. will work
9     do {
10         if(cin.fail()) {
11             cin.clear(); //reset failbit
12             cin.ignore(256, '\n'); //ignore 256 chars or until newline
13         }
14         cout << "enter int: ";
15         //cin ignores leading whitespace and reads until char not of
16         //the type specified or whitespace (newline, space, tab, etc.)
17         cin >> x;
18         cout << "value of x: " << x << endl;
19     } while(cin.fail()); //fail bit set if it doesn't get type expected
20
21     cout << "enter a float: ";
22     cin >> f;
23     cout << f << endl;
24
25     return 0;
26 }
```

In-Class Exercise #5

- How would you determine if user entered a good positive int without using `cin.clear()` and `cin.ignore()`?

```
4. ENGR
Re-attach Fullscreen Stay on top Duplicate
1 #include <iostream>
2 #include <string> //c++ strings
3 #include <cstdlib> //atoi()
4
5 using namespace std;
6
7 int main() {
8     int x;
9     float f;
10    string s; //create a string object
11    bool bad; //create a flag to indicate bad or good data
12
13    //cin and getline difference: cin stops reading at any whitespace
14    //and stays at the whitespace where it left off reading, getline
15    //reads until a newline is encountered and moves past the newline
16    cout << "enter a string: ";
17    //cin >> s; //read a string of chars from the user until whitespace
18    getline(cin, s); //read chars from the user until a newline
19    cout << "length: " << s.length() << endl;
20    cout << "First char: " << s.at(0) << endl; //0 is 1st location
21    cout << "The whole string: " << s <<endl;
22
23    do {
11,59 Top
-- INSERT -- 44,2 Bot
```

```
22
23     do {
24         bad=false; //assume the user is not going to supply bad data
25
26         cout << "enter int: ";
27         cin >> s; //read data as a string, so it will never fail.
28         //check that all the characters are 0-9 for a postivie int
29         for(int i=0; i<s.length(); i++)
30             if(!(s.at(i)>='0' && s.at(i)<='9')) {
31                 bad=true; //if a char in the string is not 0-9, not positive int
32                 break; //break out of the for loop when seeing bad data
33             }
34     } while(bad); //while the user entered bad data re-prompt them
35
36     //after we know input is good, then we can change it to integer, but
37     //atoi takes a c-style string as input, not a c++ style string
38     x=atoi(s.c_str());
39     cout << x << endl;
40
41     cout << "enter a float: ";
42     cin >> f;
43     cout << f << endl;
44
45     return 0;
46 }
```

-- INSERT --

46,2

Bot

More about **break**, **exit**, and **return**

- **break** – used with switch and loops, breaking out of the closest associated case or loop(for, while, or do while). **This statement can only occur in a loop or case**, otherwise the compiler yells!
- **return** – leave the current function, which exits the program when in the main() function. You can put this **anywhere inside any function**, otherwise the compiler yells!
- **exit()** – exit the entire program, no matter where this is encountered. You can put this **anywhere inside any function, as long as you include <cstdlib>**, otherwise the compiler yells!