PIC 10A: Week 3b

Section 1C, Spring 2016

Prof. Michael Lindstrom (TA: Eric Kim)

v1.0

Announcements

- Mini-Midterm 1 this Friday during lecture (tomorrow!)
  - Covers up to and including Monday's lecture (4/11)
  - strings will not be on the midterm
- HW1 was due on Wednesday
-

MiniMidterm 1 Topics

- Week 1: Computer hardware, compilation process
  - RAM, CPU, preprocessor, compiler, assembler, linker, etc
- Week 2: Libraries, namespaces, C++ programming, escape characters, numeric data types, binary representation
  - <iostream>, using namespace std, \n, \t, \", \', etc
  - int, long long, double, converting to/from binary/decimal
- Week 3: variables, static_cast, operators, numerical issues, user input
  - static_cast<double>(x), x++, x*=3, overflow/underflow/roundoff, getline vs cin, cin.get(), cin.ignore(), issues with mixing "cin >> x" and "getline(cin, s)"
  - Data types: char, bool

Today

- getline(), cin.ignore(), cin.get()
- strings
-

getline

```
getline(cin, string str)
```

Behavior: Asks for user input. Places **everything** you type before hitting <ENTER>, and copies it to a string object.

getline: Example

```
string mystr;
getline(cin, mystr);
cout << "mystr: " << mystr;
```

User types:
My name is Eric. It is 1:21 PM.<ENTER>

Output:

```
mystr: My name is Eric. It is 1:21 PM.
```

getline: Example

```
string mystr;
getline(cin, mystr);
cout << mystr;
```

(A)

```
string mystr;
cin >> mystr;
cout << mystr;
```

(B)

User types:
Hello world!<ENTER>

Question: What is the output of both programs?

Answer:
(A) `Hello world!`
(B) `Hello`

```
string a; string b;
cin >> a;
getline(cin, b);
cout << "a:" << a << endl;
cout << "b:" << b;
```

User inputs:
in my life<ENTER>

Question: What is the output?

Output:
```
a:in
b: my life
```

Note the space!

```
int n;
cout << "How many nickles?" << endl;
cin >> n;
cout << "What is your name?";
string myname;
getline(cin, myname);
cout << "Nb nickels: " << n;
cout << "Hi " << myname << "!";
```

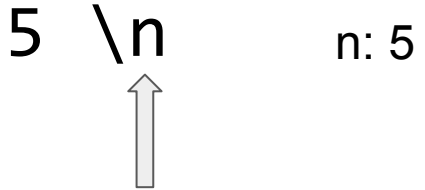
User Types:
5<ENTER>
Eric<ENTER>

Question: What is the output? Is there any weird behavior?

Answer:
Nb nickels: 5
Hi !

Warning: The code does *not* pause at the getline()!

```
int n;
cout << "How many nickles?" << endl;
cin >> n;
cout << "What is your name?";
string myname;
getline(cin, myname);
cout << "Nb nickels: " << n;
cout << "Hi " << myname << "!";
```



The newline ***stays*** in cin's buffer!

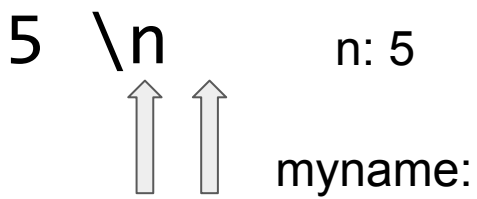
User Types:
5<ENTER>
Eric<ENTER>

Warning: The code does *not* pause at the getline()!

```
int n;
cout << "How many nickles?" << endl;
cin >> n;
cout << "What is your name?";
string myname;
getline(cin, myname);
cout << "Nb nickels: " << n;
cout << "Hi " << myname << "!";
```

User Types:
5<ENTER>
Eric<ENTER>

Warning: The code does *not* pause at the getline()!



getline() immediately returns because it found a newline!
myname is set to the empty string.
Also: getline() discards the newline in the buffer.

cin.ignore(), cin.get()

- ignore(): Discards first character in cin's buffer. If the buffer is empty, then ignore() will ask the user to type something.
- get(): Returns (and discards) first character in cin's buffer. If the buffer is empty, then get() will ask the user to type something.

```
string a; string b;
cin >> a;
cin.ignore();
getline(cin, b);
cout << "a:" << a << endl;
cout << "b:" << b;
```

User inputs:
in my life<ENTER>

Question: What is the output?

Output:
```
a:in
b:my life
```

No more space!

```
string a; string b; char c;
cin >> a;
c = cin.get();
getline(cin, b);
cout << "a:" << a << endl;
cout << "b:" << b;
cout << "c:" << c << ".\n";
```

User inputs:
in my life<ENTER>

Question: What is the output?

Output:

```
a:in
b:my life
c: .
```

Space!

```
#include <iostream>
#include <string>
int main() {
  string s;
  cin >> s;
  cout << s;
  cin.get();
  return 0;
}
```

User types:
Hi<ENTER>

Question: Does the program pause at this this line?

Answer: Nope! The newline from the previous "cin << s" gets returned from the cin.get() call.