Lecture 15: Loops

(+ a little bit of strings)

Announcements and reminders

Interview Grading (Project 1)

- **Not optional** (40 pts)
- Sign up and do by Friday 3 March
- Can earn pts even if your codes weren't working! Horrayyyy!

Submissions:

- HW 5 -- due Saturday at 6 PM

Practicum 1 -- Wednesday 5:30 - 7 PM (staggered start, don't be late, nor alarmed)

- Practicum 1 room assignments posted to Piazza:
  - 301, 303 -- ECCR 265
  - 302 -- ECCR 200
  - 304 -- ECCR 1B40
- Practice problems on Moodle -- **DO THEM. They are excellent practice problems for the practicum. That's why we call them "practice problems"**

Last time on *Intro Computing…*

- We learned about the `string` variable type!
  - … how to input strings from keyboard (`cin`)
  - … how to take subsets of strings (`str.substr(start, length)`)
  - … how to find the length of a string (`str.length()`)

Traversing a string with loops

We can loop over the characters in a string as follows:

```
void traverse(string x)
{
  for(int i=0; i < x.length(); i++)
  {
    cout << x[i] << endl;
  }
}

int main()
{
  string my_str = "ABC";
  traverse(my_str);
  return 0;
}
```

Chapter 4: Loops

1. The `while` loop
2. Problem solving: hand-tracing
3. The `for` loop
4. The `do` loop
5. Processing input
6. Problem-solving: storyboards
7. **Common loop algorithms**
8. Nested loops
9. Problem solving: solve a simpler problem first
10. Random numbers and simulations

Counting matches

Counting chars in a string:

```
int spaces = 0;
for (int i=0; i < my_str.length(); i++)
{
  if (my_str.substr(i, 1)==" ")
  {
    spaces++;
  }
}
```

Counting matches

Counting words in a user input sequence

```
int short_words = 0;
string input;
while (cin >> input)
{
   if (input.length() <= 3)
   {
      short_words++;
   }
}
```

Counting matches

Counting words in a user input sequence

```
int short_words = 0;
string input;
while (cin >> input)
{
   if (input.length() <= 3)
   {
      short_words++;
   }
}
```

This `while (cin >> input)` statement is super helpful!

- Keeps taking input from cin and doing the stuff in the `while` loop…
- … until either:

1) the user enters something that **does not conform to the type of input expected** (here, that's if the user enters something that is not a string, but that's tough…)
2) Or the **cin** encounters an **End-Of-File (EOF)** sequence. Here, we can enter:

   Windows: Ctrl+F

   Unix (Mac/Linux): Ctrl+D

Finding first location

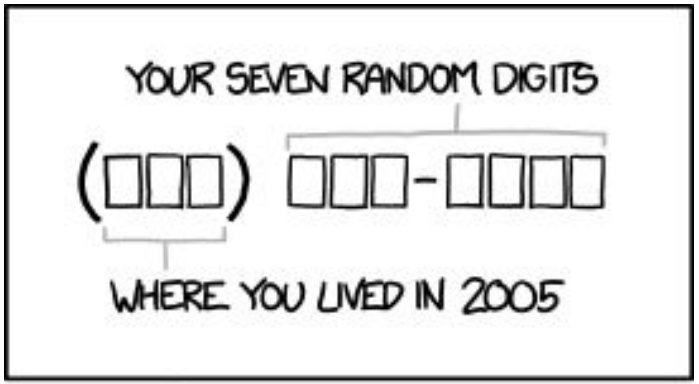
```
bool found = false;
int position = 0;
while (!found && position < my_str.length())
{
   string ch = my_str.substr(position, 1);
   if (ch == " ")
   {
      found = true;
   }
   else
   {
      position++;
   }
}
```

Finding first location… and the use of break

```
bool found = false;
int position = 0;
while (position < my_str.length())
{
   string ch = my_str.substr(position, 1);
   if (ch == " ")
   {
      found = true;
      break;
   }
   else
   {
      position++;
   }
}
```

Removing parts of a string

Example: S'pose someone enters a phone number we want to remove all spaces and hyphens from the number. How can we do this?



STRUCTURE OF A US CELL PHONE NUMBER

Removing parts of a string

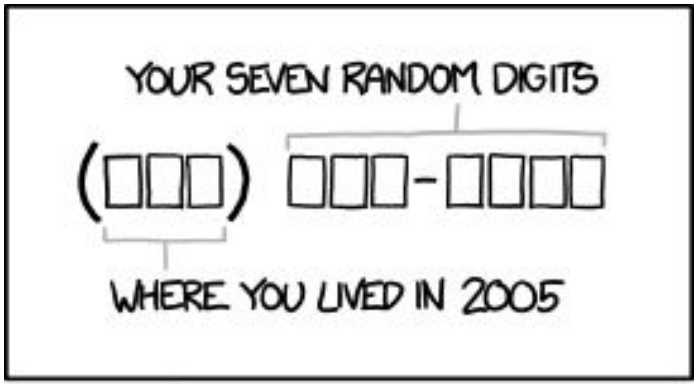
Example: S'pose someone enters a phone number we want to remove all spaces and hyphens from the number. How can we do this?

Could start with the entire phone number…

- loop over phone number, and remove any characters that are a space or hyphen
- **remove.cpp**

Could start with ***nothing***…

- loop over phone number, and ***add*** any character that is ***not*** a space or hyphen
- **remove2.cpp**



STRUCTURE OF A US CELL PHONE NUMBER

What just happened?

- We saw some common algorithms using **loops!**
  - … how to ***traverse*** a string using a loop!
  - … how to count matches in some user input!
  - … how to find the first location of something!
  - … how to remove part of a string!
    (without scissors even!)