Week 3

Announcements

- Midterm!!!
  - 10/29 (Next Thursday)
- Project 3 spec is out
  - due 11/3
  - read over it quickly and start early

Else If

```
// Clean up the code below with 'else if' statements
int income;
cin >> income;

if (income < 30)
  cout << "low"
else {
  if (income >= 30 && income < 100)
    cout << "middle"
  else {
    if (income >= 100 && income < 500)
      cout << "high"
  }
}

// *** Cleaned up version ***
int income;

if (income < 30)
  cout << "low"
else if (income < 100)
    cout << "middle"
else if (income < 500)
    cout << "high"
```

Switch Statements

```
// Clean up the code below with a 'switch' statement
int choice;
cin >> choice;

if (choice == 1)
  cout << "1" << endl;
else if (choice == 2 || choice == 3)
  cout << "2 or 3" << endl;
else if (choice == 4)
  cout << "4" << endl;
else
  cout << "idk" << endl;

// *** Cleaned up version ***
int choice;
cin >> choice;

switch (choice) {
  case 5: case 6:  // cases can be on the same line, and don't have to be in order
    break;
  case 1:
    cout << "1" << endl;
    break;
  case 2:           // no 'break' so a 2 overflows to the code inside the next case
  case 3:
    cout << "2 or 3" << endl;
    break;
  default:          // don't have to have a default case
    cout << "idk" << endl;
    break;
  case 4:
    cout << "4" << endl;
    break;
}

/*
Notes
 - can only switch on ints
    - no strings, doubles (because comparing doubles can be tricky)
 - char works because it is stored as an int
*/
```

While Loop

```
// print "red is sus" 50 times

// can do it with 50 identical lines of code
cout << "red is sus" << endl;
...

// or use a while loop
int i = 0;        // initialize iterator variable
while (i < 50) {  // check a condition
  cout << "red is sus" << endl;
  i++;            // update iterator, don't forget this/be careful
                  // it's easy to make mistakes and create infinite loops
}
```

For Loop

```
// the equivalent of the while loop above
for (int i = 0; i < 50; i++)
  cout << "red is sus" << endl;

/*
Questions

- what if we update with ++i (prefix) instead of i++ (postfix)?
    - short answer: in a for loop, they are equivalent

look at the example below to see how the two behave differently in other
situations

// Prefix increment
let prefix = 1;
console.log(++prefix); // 2
console.log(prefix); // 2

// Postfix increment
let postfix = 1;
console.log(postfix++); // 1
console.log(postfix); // 2
*/
```

Nested For Loops

```
// write a times table up to 3x3 with nested for loops
/* output
1 2 3
2 4 6
3 6 9
*/

for (int i = 1; i <= 3; i++) {
  for (int j = 1; j < 4; j++)
    cout << i * j << " ";
  cout << endl;
}
```

Char

```
// use single-quotes for char, not double quotes (which are for strings)
char letter = 'a';
// char letter2 = "b";  // WRONG

// escape sequences
char newline = '\n';
char tab = '\t';
// char badline = '/n' // WRONG, use backslashes to escape

string l = "abc\tdef\ng\n";
cout << l;
/*
prints the following:
abc    def
g

*/
```

Substrings

```
// print out how many lower case e's in the string s
string s = "eeE dog cat eel";
int sum = 0;
```

```
for (int i = 0; i != s.size(); i++) {
  char temp = s.at(i);
  // char temp = s[i];  // can do this instead of s.at(i)

  if (temp == 'e')
    sum++;
}

cout << sum << endl;

/*
Questions
 - Why didn't you do i < s.size() in the for loop?

*/

// *** Challenge problem ***
// output "yes" if 'sub' is a substring of 'word'

/*
General idea -
loop through the characters of word
loop through sub to see if the characters match
*/

string word = "anteater";
string sub = "ant"

bool flag = false;

// the stopping condition is so i+j doesn't trigger an out-of-bounds error
for (int i = 0; i != word.size() - sub.size(); i++) {

  for (int j = 0; j != sub.size(); j++) {

    char c1 = word.at(i+j); // make sure this doesn't go past the last char in word
    char c2 = sub.at(j);

    if (c1 != c2) {
      flag = false;         // if any characters don't match up, break and set flag
      break;
    }
    flag = true;            // if all characters match up, flag will be true at the
                            // end of the 2nd for-loop
  }

  if (flag)
    break;                  // if we've found a match, break out of 1st for-loop
                            // no need to keep looking
}

if (flag)
  cout << "yes";
```