C++ IV

Functions and Arrays

Declaring and defining functions

- Functions must be declared or defined before they are used
- They can be declared before being defined
- Implement them like you would in Java

```
1  #include <iostream>
2
3  using namespace std;
4
5  int mult ( int x, int y );
6
7  int main()
8  {
9    int x;
10   int y;
11
12   cout << "Input two numbers: ";
13   cin >> x >> y;
14   cin.ignore();
15   cout << "Their product is "
16   << mult ( x, y ) << "\n";
17   cin.get();
18 }
19
20 int mult ( int x, int y )
21 {
22   return x * y;
23 }
```

Syntax:

```
return-type function_name(arg_type arg1, ..., arg_type argN);
```

Arrays

Creating an array syntax:

```
type arrayName[number];
```

Assigning a value to an array:

```
arrayName[indx] = someVal;
```

```
#include <iostream>
using namespace std;

int main(){
  int x;
  int y;
  // A 2D array
  int array[8][8];

  for ( x = 0; x < 8; x++ ) {
    for ( y = 0; y < 8; y++ )
      // Set each element to a value
      array[x][y] = x * y;
  }
  cout<<"Array Indices:\n";
  for ( x = 0; x < 8;x++ ) {
    for ( y = 0; y < 8; y++ )
      cout<<"["<<x<<"]["<<y<<"]="
      << array[x][y] <<" ";
    cout<<"\n";
  }
  return 0;
}
```

Strings

- Two kinds of strings, cstrings (char arrays) and c++ strings: we will focus on the latter
- strings are part of the standard namespace
- declaring a string: `string myString;` or `string myString("ask");`
- Strings can be concatenated as in Java
- They can be compared with ==

```
#include <iostream>
using namespace std;

int main(){

    //concatenation
    string my_string1 = "a string";
    string my_string2 = " is this";
    string my_string3 = my_string1 + my_string2;

    // Will ouput "a string is this"
    cout<<my_string3<<endl;

    string passwd;
    cout<<"Enter Password"<<endl;
    getline(cin, passwd, '\n');
    if(passwd == "xyzzy")
    {
        cout<<"Access allowed";
    }

}
```

Strings Contd

- Strings have both a length() and size() function
- individual chars: myString[idx];
- Strings can be searched, substrings can be taken (see code to right)

```
#include <iostream>
using namespace std;

int main(){
    string my_string("starting value");
    string my_string1 = "ten chars.";
    int len = my_string1.length(); // or .size();

    for(int i = 0; i < my_string.length(); i++){
        cout<<my_string[i];
    }
    cout<<endl;

    string input;
    int cat_appearances = 0;
    cout<<"Enter something"<<endl;

    getline(cin, input, '\n');

    for(int j = input.find("cat", 0); j != string::npos; j = input.find("cat", j))
        cat_appearances++;
        j++;  // Move past the last discovered instance to avoid finding same
              // string
    }

    cout<<cat_appearances<<endl;

    return 0;
}
```

Strings Contd

- Strings are mutable
- Strings can be modified as seen in the code to the right
- Passing strings into functions results in their being copied, though the copy might be delayed

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      string my_removal = "remove aaa";
6      my_removal.erase(7, 3); // erases aaa
7
8      string my_string = "ade";
9      my_string.insert(1, "bc");
10
11     // my_string is now "abcde"
12     cout<<my_string<<endl;
13
14     return 0;
15 }
```

A Fun Exercise (just for fun)

- Write a function that takes a string returns a bool that checks if the string is a palindrome with the following quirk: all c and d characters do not count, but all others do.
- Try to do it in place, without using an extra data structure.