Class 26

Prefix and Postfix Increment, Break, Continue
Arguments to Main

Final Exam

- Monday, May 22 from 6:15 pm to 8:15 pm
- Exam will be held in the RO-230
- If you miss the final exam, you will receive a grade of WU per CUNY policy.
- Six questions
- Exam is cumulative across entire semester

1. Title line
2. Code output
3. Short code blocks OR debugging code
4. Complete C++ program involving strings and string methods
5. Complete C++ program involving files I/O and patterns
6. Complete C++ program involving characters and arrays

(any of 4-6 may also involve functions)

Example 0: Generate a random letter

Recall that there is a correspondence between integers and characters:

- ASCII 65 to 90 = ‘A’ to ‘Z’ (upper case)
- ASCII 97 to 122 = ‘a’ to ‘z’ (lower case)

Generate a random int in the desired range, and cast as a char

- char x = rand() % ('Z' - 'A' + 1) + 'A’; // generates random upper case letter
- cout << x << endl;
- cout << (char)(rand() % ('z' - 'a' + 1) + 'a’); // prints random lower case letter

Prefix and postfix increment/decrement

Given int i

- i++ - this is a postfix increment operation. The expression is evaluated first using the original value of the variable. After evaluating the expression, the variable is incremented.
- ++i - this is a prefix increment operation. The variable is incremented first and then the expression is evaluated using the new value of the variable

Example 1: Pre-increment & Post-increment

```
int main(){
   int i = 5, j = 5;
   cout << ++i << endl; // prints 6
   cout << i << endl; // prints 6
   cout << j++ << endl; // prints 5
   cout << j << endl; // prints 6
   return 0;
}
```

Another example:

```
int main(){
   int i = 5, j, k;
   j = i++;
   k = ++i;
   cout << j << endl; // j stores 5
   cout << k << endl; // k stores 7
   return 0;
}
```

Example 2: Loop control: break, continue

Inside any loop, you can set a condition to either “jump out” of the loop, or skip a particular iteration of the loop and continue with the next iteration

```
for(int i = 1; i <= 10; i++){
    if(i % 3 == 0) continue;
    cout << i << endl;
}
```

```
for(int i = 1; i <= 10; i++){
    if(i == 8) break;
    cout << i << endl;
}
```

Arguments to main

- Alternative title line to main
- We can set up a main program to work with arguments
- These are called command-line arguments

mars> ./a.out file1 file2

Main title line for command-line arguments

- Old title line:
  - int main()
- New title line:
  - int main(int argc, char *argv[])
  - This version of the main title line allows main to take command-line arguments
  - argc is the number of inputs
  - argv is an array showing the inputs
    - argv is an array of c-strings

Example 3: Arguments to main

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

int main(int argc, char *argv[]){     // Command line arguments are stored in an array called argv

  for(int i = 0; i < argc; i++){

    cout << argv[i] << endl;

  }

  return 0;

}
```

argc, argv

mars>./a.out file1 file2

int main(int argc, char *argv[])

argc: how many things did the user type?

argv: what did the user type?

Another example:

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

int main(int argc, char *argv[]){
    ifstream f1;
    ofstream f2;
    f1.open(argv[1]);  // connect f1 to command-line argument
    f2.open(argv[2]);  // connect f2 to command-line argument
    char x = f1.get(); // get the first character from f1
    while(!f1.eof()){
        f2 << x;       // copy char x to f2
        x = f1.get();  // get the next character; assign to x
    }
    f1.close();
    f2.close();
    return 0;
}
```