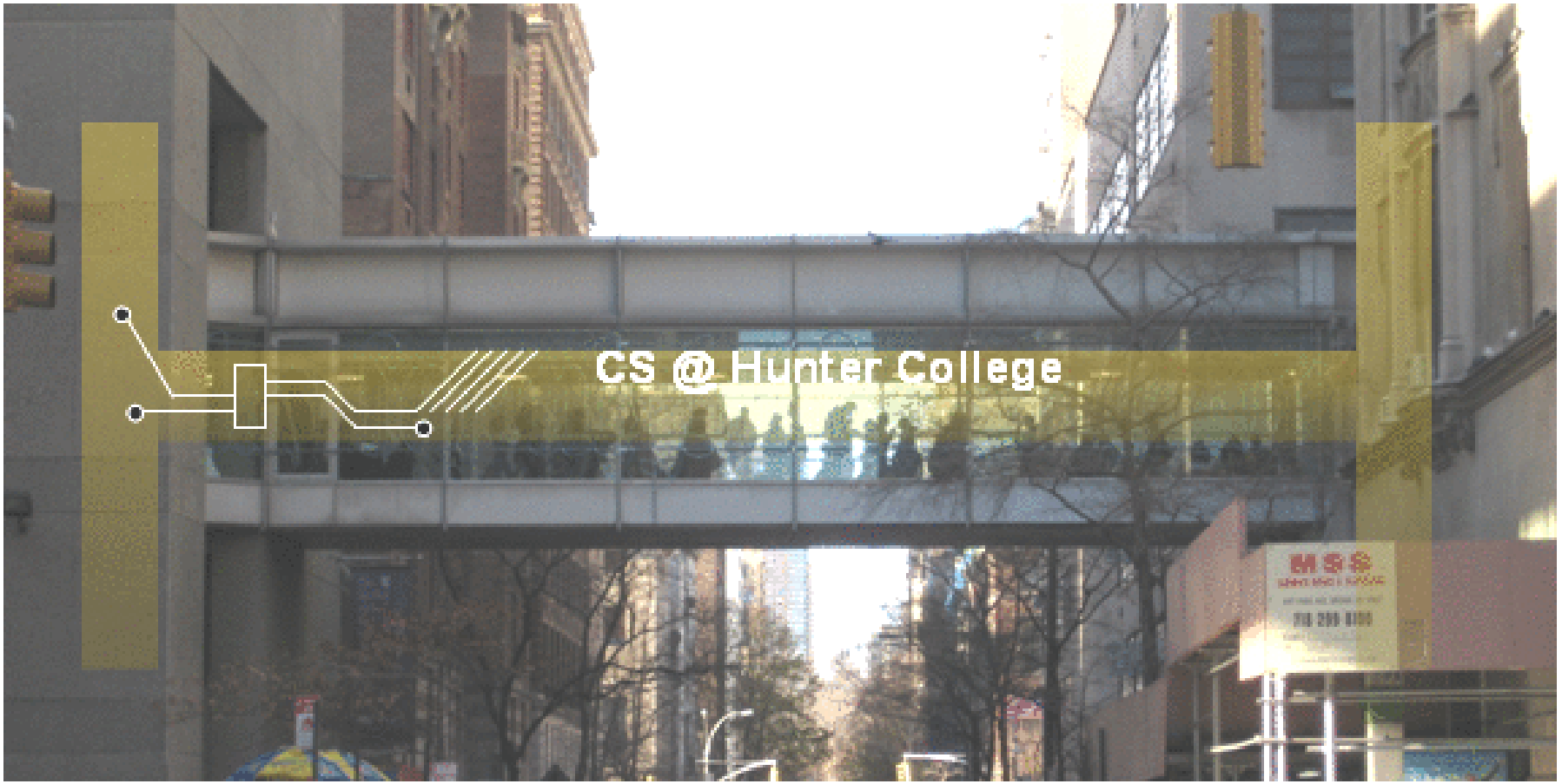
CSCI 127: Introduction to Computer Science



hunter.cuny.edu/csci

Recap: weMIPS

- Machine language is a programming language that is lower-level on the levels of abstraction
- weMIPS is an educational tool that is a simplified version of a real assembly language called MIPS
- We run our weMIPS programs using an online emulator

Python to weMIPS

Consider the following Python program:

```
s0 = 0 + 1
s1 = s0 + s0
s2 = s1 + s1
s3 = s2 + s2
```

Python to weMIPS

Recall the following weMIPS program:

```
ADDI $s0, $zero, 1    #s0 = 0 + 1
ADD $s1, $s0, $s0     #s1 = s0 + s0
ADD $s2, $s1, $s1     #s2 = s1 + s1
ADD $s3, $s2, $s2     #s3 = s2 + s2
```

Intro C++

- Intro C++ syntax
- Translating from Python to C++
- Input/Output (I/O):
  - `cin >>`
  - `cout <<`
- Definite loops:

```
for (int i = 0; i < 10; i++) {
    ...
}
```

OnlineGDB

- Running your C++ code locally requires additional setup (see Lab 12)
- You can also use OnlineGDB

Your first C++ program

- The commands we want to be run should go inside of the main function but before the call to `return 0`
- Notice the opening and closing curly braces
- In C++, curly braces denote the beginning and end of a block of code
- Comments are denoted with two forward slashes: `//comment`
- Statements are ended with semicolons

```
int main() {
    //All commands go here

    return 0;
}
```

Hello, World! (C++)

- To print values in C++, we need to import a library called `iostream`
- From the `iostream` library, we can use the `cout` function
- This works similarly to `print()` in Python but has different syntax

```
#include <iostream>

int main() {
    std::cout << "Hello, World!";
    return 0;
}
```

Hello, World! (C++)

- A function that comes from a library needs to have `std::` in front of it
- We can avoid that by including the line `using namespace std;` at the bottom of the library imports

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

int main() {
    cout << "Hello, World!";
    return 0;
}
```

Hello, World! (C++)

- Let's try to print two times...
- It works, but the values are printed right next to each other: `Hello, World!Hello, World!`

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

int main() {
    cout << "Hello, World!";
    cout << "Hello, World!";
    return 0;
}
```

Hello, World! (C++)

- We can use the newline character "\n" to make sure the next value printed appears on the next line

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

int main() {
    cout << "Hello, World!\n";
    cout << "Hello, World!\n";
    return 0;
}
```

Definite Loops in Python

```
for i in range(10):
    print("Hello, World!")
```

- What are the start, stop, and step values for this call to `range()`?
- Start is 0, stop is 10, step is 1
- Though we don't use the values of `i`, we know that 10 numbers are generated, so the loop will repeat 10 times (i.e., once for each number generated)

Definite Loops in C++

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

int main() {
//  for(int i = start; i < stop; i = i + step) {...}
    for(int i = 0; i < 10; i = i + 1) {
        cout << "Hello, World!\n";
    }
}
```

Getting user input in C++

- The `iostream` library also defines the `cin` function
- We can use this to get user input

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

int main() {
    cout << "Enter a number: ";
    int stop;
    cin >> stop;

    cout << "You entered: " << stop << "\n";
}
```

Getting user input in C++

- We should first prompt the user with a message from `cout` so they know what to enter
- We can then grab that value using `cin`, but we need a variable to store the user's input
- C++ is a strongly typed language, meaning we have to provide the variable's type when we create it; we are asking the user for a number, so let's store the variable as an integer type

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

int main() {
    cout << "Enter a number: ";
    int stop;
    cin >> stop;
    cout << "You entered: " << stop << "\n";
}
```

Putting it together

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

int main() {
    cout << "Enter a number: ";
    int stop;
    cin >> stop;

    for(int i = 0; i < stop; i = i + 1) {
        cout << "Hello, World!\n";
    }
    return 0;
}
```

Putting it together

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

int main() {
    cout << "Enter a number: ";
    int stop;
    cin >> stop;

    for(int i = 0; i < stop; i++) {
        cout << "*";
    }
    cout << "\n";

    return 0;
}
```

Final Exam

- Early exam: Weds December 11 at 9 AM in room 1001G HN
- If you take the early exam, you cannot take the regular exam
- Regular exam: MONDAY Decemeber 16 at 9 AM in Assembly Hall
- If you do not take either the early or regular final exam, you automatically get a grade of WU
- If you want your WU to become a letter grade, you have to take the late exam in January 2025