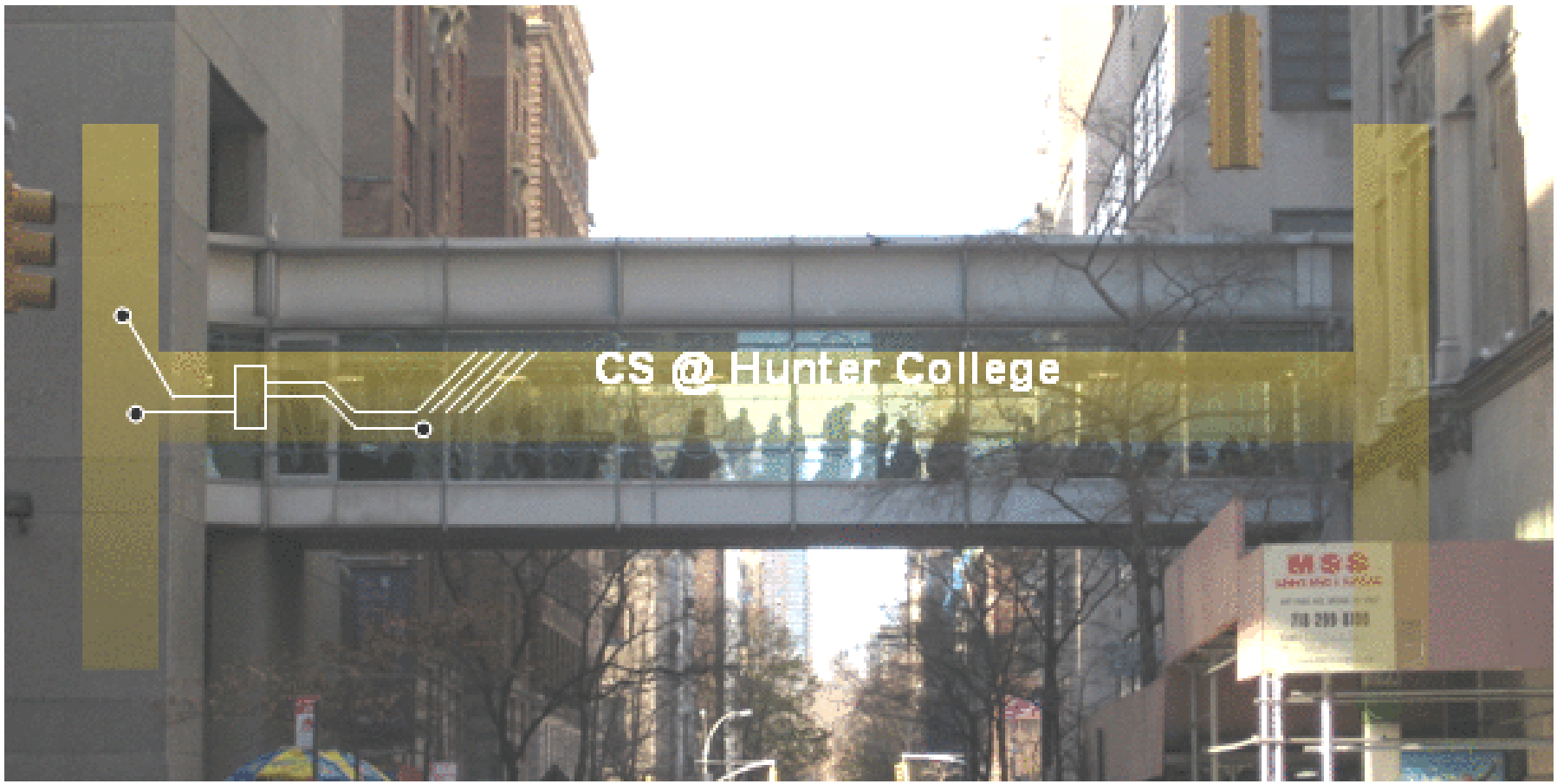
CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

Teacher Evaluation

- Please take a moment to fill out the **Teacher Evaluations**

Teacher Evaluation

- Please take a moment to fill out the **Teacher Evaluations**
- Your chance to provide feedback!

Teacher Evaluation

- Please take a moment to fill out the **Teacher Evaluations**
- Your chance to provide feedback!
- Responses are **completely anonymous.**

Teacher Evaluation

- Please take a moment to fill out the **Teacher Evaluations**
- Your chance to provide feedback!
- Responses are **completely anonymous.**
- Smartphone: www.hunter.cuny.edu/mobilete

Teacher Evaluation

- Please take a moment to fill out the **Teacher Evaluations**
- Your chance to provide feedback!
- Responses are **completely anonymous.**
- Smartphone: www.hunter.cuny.edu/mobilete
- Computer: www.hunter.cuny.edu/te

Announcements

- **Final Exam May 22 at 9-11 AM**

Announcements

- **Final Exam May 22 at 9-11 AM**
- **Seating assignment will be available on Blackboard/My Grades by this week**
- Next **Tuesday, May 16, we will have a Mock Exam**

Announcements

- **Final Exam May 22 at 9-11 AM**
- **Seating assignment will be available on Blackboard/My Grades by this week**
- Next **Tuesday, May 16, we will have a Mock Exam**
  - Room 118 Hunter North (Assembly Hall), ground floor of the North Building

Announcements

- **Final Exam May 22 at 9-11 AM**
- **Seating assignment will be available on Blackboard/My Grades by this week**
- Next **Tuesday, May 16, we will have a Mock Exam**
  - Room 118 Hunter North (Assembly Hall), ground floor of the North Building
  - Only 1 hr 15 mins for the Mock, 2 hours for the real exam.

Announcements

- **Final Exam May 22 at 9-11 AM**
- **Seating assignment will be available on Blackboard/My Grades by this week**
- Next **Tuesday, May 16, we will have a Mock Exam**
  - Room 118 Hunter North (Assembly Hall), ground floor of the North Building
  - Only 1 hr 15 mins for the Mock, 2 hours for the real exam.
  - Just a practice run and it will NOT be graded (answer keys will be posted).
  - However, the mock will have the same logistics and question format as the real final exam.

Frequently Asked Questions

- What's the best way to study for the final exam?

Frequently Asked Questions

- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.

Frequently Asked Questions

- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are online. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update your note sheet, and repeat.

Frequently Asked Questions

- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are online. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update your note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit?

Frequently Asked Questions

- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are online. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update your note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.

Frequently Asked Questions

- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are online. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update your note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?

Frequently Asked Questions

- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are online. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update your note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?
  First: it gives unfair advantage & is immoral.

Frequently Asked Questions

- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are online. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update your note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?
  First: it gives unfair advantage & is immoral.
  Second: it degrades the quality of our students.

Frequently Asked Questions

- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are online. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update your note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?
  First: it gives unfair advantage & is immoral.
  Second: it degrades the quality of our students.
  Third: it's a standard question on faculty references.

Frequently Asked Questions

- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are online. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update your note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?
  First: it gives unfair advantage & is immoral.
  Second: it degrades the quality of our students.
  Third: it's a standard question on faculty references.
  Industry & graduate schools hate it: don't want someone who falsifies work.

Announcements

For those of you taking CSCI 135 next semester (Fall 2023):

- On the first day of CSCI 135, you will have a graded quiz on the C++ topics covered in CSCI 127.
- This includes all examples covered in the labs and lectures as well as the last 8 homework problems.
- You will have to remember and review all this material during the Summer so you are prepared for the first day of class.

Today's Topics

```cpp
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Recap: I/O & Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Guest: Prof. Ahearn, Geography

Today's Topics

```cpp
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- **Recap: I/O & Definite Loops in C++**
- Conditionals in C++
- Indefinite Loops in C++
- Guest: Prof. Ahearn, Geography

Recap: Basic Form & I/O in C++

```
//C++ program demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

Recap: Basic Form & I/O in C++

- Efficient for systems programming.

```cpp
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

Recap: Basic Form & I/O in C++

- Efficient for systems programming.
- Programs are organized in functions.

```cpp
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

Recap: Basic Form & I/O in C++

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables:

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

Recap: Basic Form & I/O in C++

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

Recap: Basic Form & I/O in C++

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

Recap: Basic Form & I/O in C++

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

Recap: Basic Form & I/O in C++

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print:

Recap: Basic Form & I/O in C++

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print: `cout << "Hello!!";`

Recap: Basic Form & I/O in C++

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print: `cout << "Hello!!";`
- To get input:

Recap: Basic Form & I/O in C++

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print: `cout << "Hello!!";`
- To get input: `cin >> num;`

Recap: Basic Form & I/O in C++

```cpp
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print: `cout << "Hello!!";`
- To get input: `cin >> num;`
- To use those I/O functions:

Recap: Basic Form & I/O in C++

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print: `cout << "Hello!!";`
- To get input: `cin >> num;`
- To use those I/O functions:
  `#include <iostream>`
  `using namespace std;`

Recap: Basic Form & I/O in C++

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print: `cout << "Hello!!";`
- To get input: `cin >> num;`
- To use those I/O functions:
  `#include <iostream>`
  `using namespace std;`
- Definite loops:

Recap: Basic Form & I/O in C++

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print: `cout << "Hello!!";`
- To get input: `cin >> num;`
- To use those I/O functions:
  `#include <iostream>`
  `using namespace std;`
- Definite loops:
  `for (i = 0; i < 10; i++) {...}`

Recap: Basic Form & I/O in C++

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print: `cout << "Hello!!";`
- To get input: `cin >> num;`
- To use those I/O functions:
  `#include <iostream>`
  `using namespace std;`
- Definite loops:
  `for (i = 0; i < 10; i++) {...}`
- Blocks of code uses '{' and '}'.

Recap: Basic Form & I/O in C++

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print: `cout << "Hello!!";`
- To get input: `cin >> num;`
- To use those I/O functions:
  `#include <iostream>`
  `using namespace std;`
- Definite loops:
  `for (i = 0; i < 10; i++) {...}`
- Blocks of code uses '{' and '}'.
- Commands generally end in ';'.

Today's Topics

```cpp
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Recap: I/O & Definite Loops in C++
- **Conditionals in C++**
- Indefinite Loops in C++
- Guest: Prof. Ahearn, Geography

Challenge:

Predict what the following pieces of code will do:

```cpp
//Demonstrates conditionals
#include <iostream>
using namespace std;

int main ()
{
    int yearBorn;
    cout << "Enter year born: ";
    cin >> yearBorn;
    if (yearBorn < 1946)
    {
        cout << "Greatest Generation";
    }
    else if (yearBorn <= 1964)
    {
        cout << "Baby Boomer";
    }
    else if (yearBorn <= 1984)
    {
        cout << "Generation X";
    }
    else if (yearBorn <= 2004)
    {
        cout << "Millennial";
    }
    else
    {
        cout << "TBD";
    }

    return 0;
```

```cpp
using namespace std;

int main ()
{
    string conditions = "blowing snow";
    int winds = 100;
    float visibility = 0.2;

    if ( ( (winds > 35) && (visibility < 0.25) )
         ( (conditions == "blowing snow") ||
           (conditions == "heavy snow") ) )
        cout << "Blizzard!\n";

    string origin = "South Pacific";

    if (winds > 74)
        cout << "Major storm, called a ";
    if ((origin == "Indian Ocean")
        ||(origin == "South Pacific"))
        cout << "cyclone.\n";
    else if (origin == "North Pacific")
        cout << "typhoon.\n";
    else
        cout << "hurricane.\n";
```

Conditionals

```
//Demonstrates conditionals
#include <iostream>
using namespace std;

int main ()
{
    int yearBorn;
    cout << "Enter year born: ";
    cin >> yearBorn;
    if (yearBorn < 1946)
    {
        cout << "Greatest Generation";
    }
    else if (yearBorn <= 1964)
    {
        cout << "Baby Boomer";
    }
    else if (yearBorn <= 1984)
    {
        cout << "Generation X";
    }
    else if (yearBorn <= 2004)
    {
        cout << "Millennial";
    }
    else
    {
        cout << "TBD";
    }

    return 0;
}
```

General format:

```
if ( logical expression )
{
      command1;
      ...
}
else if ( logical expression )
{
      command1;
      ...
}
else
{
      command1;
      ...
}
```

Logical Operators in C++

Very similar, just different names: `&&`, `||`, and `!`:

Logical Operators in C++

Very similar, just different names: `&&`, `||`, and `!`:

and (`&&`)

in1		in2	*returns:*
`False`	`&&`	`False`	`False`
`False`	`&&`	`True`	`False`
`True`	`&&`	`False`	`False`
`True`	`&&`	`True`	`True`

Logical Operators in C++

Very similar, just different names: `&&`, `||`, and `!`:

and (`&&`)

in1		in2	*returns:*
`False`	`&&`	`False`	`False`
`False`	`&&`	`True`	`False`
`True`	`&&`	`False`	`False`
`True`	`&&`	`True`	`True`

or (`||`)

in1		in2	*returns:*
`False`	`||`	`False`	`False`
`False`	`||`	`True`	`True`
`True`	`||`	`False`	`True`
`True`	`||`	`True`	`True`

Logical Operators in C++

Very similar, just different names: `&&`, `||`, and `!`:

and (`&&`)

in1		in2	*returns:*
`False`	`&&`	`False`	`False`
`False`	`&&`	`True`	`False`
`True`	`&&`	`False`	`False`
`True`	`&&`	`True`	`True`

or (`||`)

in1		in2	*returns:*
`False`	`||`	`False`	`False`
`False`	`||`	`True`	`True`
`True`	`||`	`False`	`True`
`True`	`||`	`True`	`True`

not (`!`)

	in1	*returns:*
`!`	`False`	`True`
`!`	`True`	`False`

Lecture Slip

- Write a complete C++ program that prompts the user to enter a time (in 24-hour format) and prints the time of day: morning, afternoon, or evening.
- Assume that afternoon is any time after 12 P.M. (1200), and that the evening is any time after 6 P.M. (1800).

Today's Topics

```cpp
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Recap: I/O & Definite Loops in C++
- Conditionals in C++
- **Indefinite Loops in C++**
- Guest: Prof. Ahearn, Geography

Challenge: predict what the code will do

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

int main ()
{
  int population = 100;
  int year = 0;
  cout << "Year\tPopulation\n";
  while (population < 1000)
  {
      cout << year << "\t" << population << "\n";
      population = population * 2;
      year++;
  }
  return 0;
}
```

C++ Demo

```cpp
///While Growth Example
#include <iostream>
using namespace std;

int main ()
{
  int population = 100;
  int year = 0;
  cout << "Year\tPopulation\n";
  while(population < 1000)
  {
    cout << year << "\t\t" << population << "\n";
    population = population * 2;
    year++;
  }
  return 0;
}
```

(Demo with `onlinegdb`)

Indefinite Loops: `while`

```
///While Growth Example
#include <iostream>
using namespace std;

int main ()
{
  int population = 100;
  int year = 0;
  cout << "Year\tPopulation\n";
  while(population < 1000)
  {
    cout << year << "\t\t" << population << "\n";
    population = population * 2;
    year++;
  }
  return 0;
}
```

General format:

`while` (*logical expression*)
{

command1;
command2;
command3;
...

}

Challenge: predict what the code does

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

int main ()
{
  int num;
  cout << "Enter an even number: ";
  cin >> num;
  while (num % 2 != 0)
  {
      cout << "\nThat's odd!\n";
      cout << "Enter an even number: ";
      cin >> num;
  }
  cout << "You entered: " << num << ".\n";
  return 0;
}
```

C++ Demo

```cpp
//Demonstrates loops
#include <iostream>
using namespace std;

int main ()
{
  int num;
  cout << "Enter an even number: ";
  cin >> num;
  while (num % 2 != 0)
  {
      cout << "\nThat's odd!\n";
      cout << "Enter an even number: ";
      cin >> num;
  }
  cout << "You entered: "
       << num << ".\n";
  return 0;
}
```

(Demo with onlinegdb)

Indefinite Loops: `while`

```
//Demonstrates loops
#include <iostream>
using namespace std;

int main ()
{
  int num;
  cout << "Enter an even number: ";
  cin >> num;
  while (num % 2 != 0)
  {
      cout << "\nThat's odd!\n";
      cout << "Enter an even number: ";
      cin >> num;
  }
  cout << "You entered: "
       << num << ".\n";
  return 0;
}
```

General format:

`while` (*logical expression*)
{

command1;
command2;
command3;
...

}

Challenge: predict what the code will do

```
//Demonstrates do-while loops
#include <iostream>
using namespace std;

int main ()
{
  int num;
  do
  {
      cout << "Enter an even number: ";
      cin >> num;
  } while (num % 2 != 0);

  cout << "You entered: " << num << ".\n";
  return 0;
}
```

C++ Demo:

```cpp
//Demonstrates do-while loops
#include <iostream>
using namespace std;

int main ()
{
  int num;
  do
  {
      cout << "Enter an even number: ";
      cin >> num;
  } while (num % 2 != 0);

  cout << "You entered: "
       << num << ".\n";
  return 0;
}
```

(Demo with `onlinegdb`)

Indefinite Loops: `do-while`

```
//Demonstrates do-while loops
#include <iostream>
using namespace std;

int main ()
{
  int num;
  do
  {
      cout << "Enter an even number: ";
      cin >> num;
  } while (num % 2 != 0);

  cout << "You entered: "
       << num << ".\n";
  return 0;
}
```

General format:

```
do
{
      command1;
      command2;
      command3;
      ...
} while ( logical expression );
```

Today's Topics

```cpp
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

- Recap: I/O & Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- **Guest: Prof. Ahearn, Geography**

Weekly Reminders!

Before the next lecture, don't forget to:

- Work on this week's Online Lab

Weekly Reminders!

Before the next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001G Hunter North

Weekly Reminders!

Before the next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001G Hunter North
- Submit this week's programming assignments

Weekly Reminders!

Before the next lecture, don't forget to:

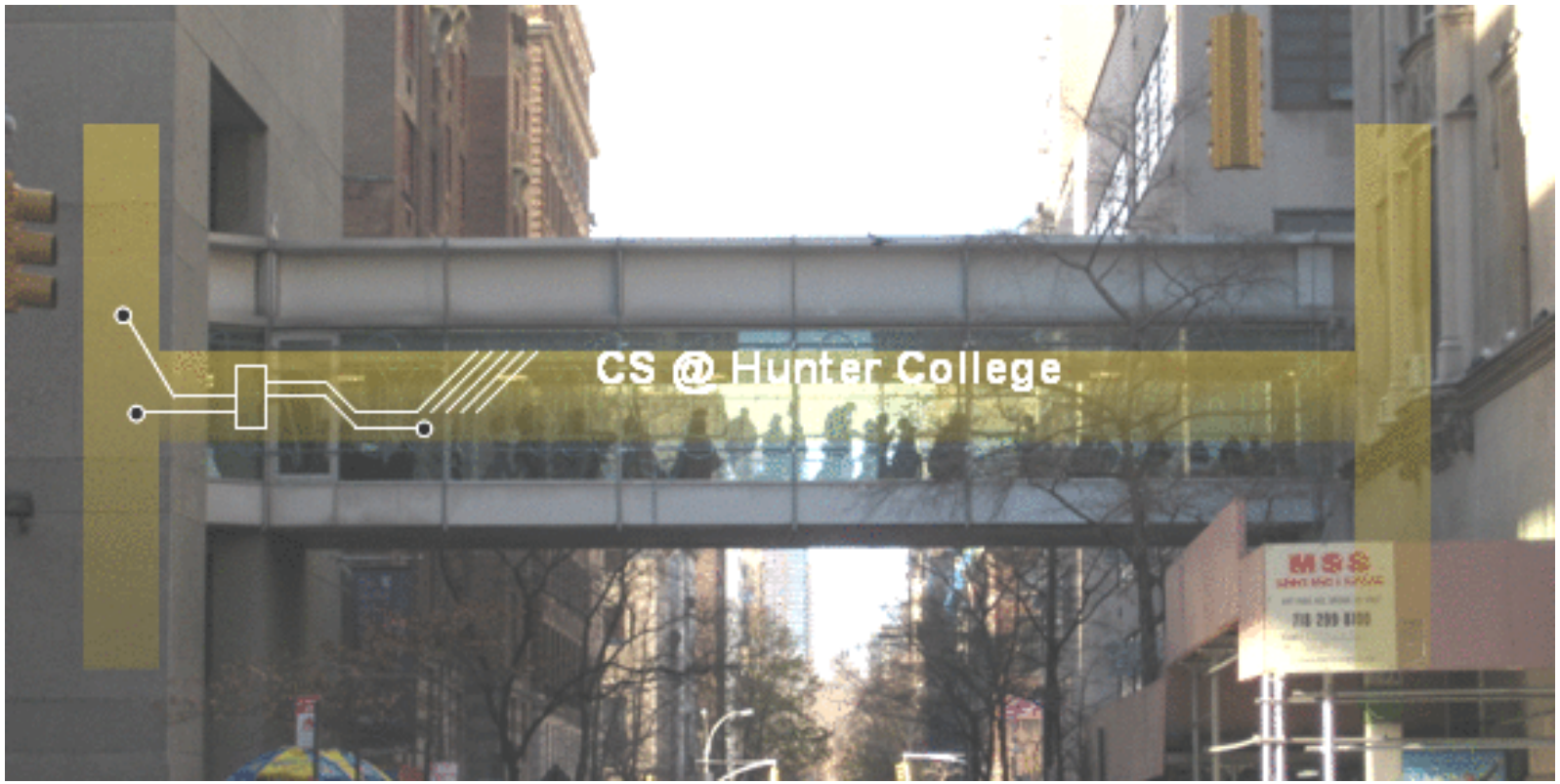
- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001G Hunter North
- Submit this week's programming assignments
- If you need help, schedule an appointment for Tutoring in lab 1001G

Weekly Reminders!

Before the next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001G Hunter North
- Submit this week's programming assignments
- If you need help, schedule an appointment for Tutoring in lab 1001G
- Take the Lecture Preview on Blackboard on Monday (or no later than 10:15am on Tuesday)

Lecture Slips & Writing Boards



- Hand your lecture slip to a UTA.
- Return writing boards as you leave.