CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

< □ > < □ > < □ > < □ > < □ >

CSci 127 (Hunter)

Lecture 13

3 7 May 2019 1 / 36

990

Announcements

Mock exam next week. Final exam: 2 weeks!



999

Announcements



- Mock exam next week.
 Final exam: 2 weeks!
- We end lecture with a survey of computing research and tech in NYC.

Today: Citi Bike's Bike Angels: Collin Waldroch

イロト イポト イヨト イヨト

7 May 2019 2 / 36

Announcements



- Mock exam next week. Final exam: 2 weeks!
- We end lecture with a survey of computing research and tech in NYC.

Today: Citi Bike's Bike Angels: Collin Waldroch

• 12:30pm: Informal Q&A with Collin in 631 Hunter East (inside library).

イロト イロト イヨト イ

Today's Topics

```
//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++
- ${\scriptstyle \bullet }$ Indefinite Loops in C++
- Recap: C++ & Python
- CS Survey

Sac

イロト 不得 トイヨト イヨト 二日

Today's Topics

```
//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++
- Recap: C++ & Python
- CS Survey

200

イロト 不得 トイヨト イヨト 二日

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;
int main ()
£
  float kg, lbs;
  cout << "Enter kg: ";</pre>
  cin >> kq;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}
```

E SQC

• Efficient for systems programming.

```
//Another C++ program, demonstrating I/O & arithmetic
#include ciostreams
using namespace std;
int main O
{ float tg, lbs;
cout << "Enter kg; ";
int >> kg; ".2;
cout << endl << "Lbs; " << lbs << "\n\n";
}
enturn 0;
}
```

イロト 不良 トイヨト イヨト ヨー のくや

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

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cisteream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "Enter kg: ";
  cin > kg; 2.2;
  coat << endl << "Lbs: " << lbs << "\n\n";
  return 0;
```

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

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cisteream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "Enter kg: ";
  cin > kg; 2.2;
  coat << endl << "Lbs: " << lbs << "\n\n";
  return 0;
```

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

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cisteream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "Enter kg: ";
  cin > kg; 2.2;
  coat << endl << "Lbs: " << lbs << "\n\n";
  return 0;
```

200

イロト 不得 トイヨト イヨト 二日

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

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cin > kg;
  cin > kg;
  coat << end! << "lbs: " << lbs << "\n\n";
  return 0;
```

200

- 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
minclude cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cin > kg;
  cin > kg;
  coat << end! << "lbs: " << lbs << "\n\n";
  return 0;
```

Sac

<pre>//Another C++ program, demonstrating I/O & a #include <iostream> using namespace std;</iostream></pre>	rithmetic
int main ()	
float ka, lbs;	
cout << "Enter kg: ";	
cin >> kg;	
lbs = kg * 2.2;	
<pre>cout << endl << "Lbs: " << lbs << "\n\n"; return 0:</pre>	
3	

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

```
int, float, char, ...
```

To print:

//Another C++ program, demonstrating I/O & arithmetic #include <iostream> using namespace std;</iostream>
int main ()
<pre>{ float kg, lbs; }</pre>
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!!";

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

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

- 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:

E 990

<pre>//Another C++ program, demonstrating I/O & arithmet #include <iostream> using namespace std;</iostream></pre>
<pre>int moin O { for the last cout << "Enter Ng: "; cin >> kg; lbs = kg * 2.2; cout << endl << "Lbs: " << lbs << "\n\n"; } return 0; }</pre>

- 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;

Sac

<pre>//Another C++ program, demonstrating 1/0 & arithme #include <iostream> using namespace std;</iostream></pre>
<pre>int main O { fout kg, lbs; cout << "Enter kg: "; cin >> kg; lbs = kg # 2.2; cout << endl << "Lbs: " << lbs << "\n\n"; feature 0; }</pre>

- 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:

200

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

3

- 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;

<pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre>
<pre>int main () { for kg, lbs; cont << "Enter kg: "; cin >> kg; lbs = kg * 2.2; cont << endl << "Lbs: " << lbs << "\n\n"; return 0;</pre>

- 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:

<pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre>
<pre>int main O { forck kg, bs; cont <= "finer kg; "; cin >> "gi; bs = kg = 2.2; cont << "cont << "libs: " << loss << "\n\n"; return @; </pre>
3

- 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++) {...}</pre>

<pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre>
int main () {
float kg, lbs;
cout << "Enter kg: ";
cin >> kg;
lbs = kg * 2.2;
cout and "the: " the "\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++) {...}</pre>
- Blocks of code uses '{' and '}'.

<pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre>
int main () {
float kg, lbs;
cout << "Enter kg: ";
cin >> kg;
lbs = kg * 2.2;
cout and "the: " the "\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++) $\{\ldots\}$

- Blocks of code uses '{' and '}'.
- Commands generally end in ';'.

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

Today's Topics

```
//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++

- ${\scriptstyle \bullet }$ Indefinite Loops in C++
- Recap: C++ & Python
- CS Survey

990

イロト 不得 トイヨト イヨト 二日

In Pairs or Triples:

Predict what the following pieces of code will do:

```
//Demonstrates conditionals
#include <iostream>
using namespace std:
int main ()
    int yearBorn;
    cout << "Enter year born: ";</pre>
    cin >> yearBorn:
    if (yearBorn < 1946)
        cout << "Greatest Generation";</pre>
    else if (yearBorn <= 1964)
    £
        cout << "Baby Boomer":
    else if (yearBorn <= 1984)
        cout << "Generation X";</pre>
    else if (vearBorn <= 2004)
        cout << "Millennial":</pre>
    }
    else
        cout << "TBD":
    return 0:
```

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

イロト イポト イヨト イヨト

```
CSci 127 (Hunter)
```

Lecture 13

3 7 May 2019 8 / 36

Sac

C++ Demo

```
//Demonstrates conditionals
#include <iostream>
using namespace std:
int main ()
£
    int yearBorn:
    cout << "Enter year born: ";</pre>
    cin >> yearBorn;
    if (yearBorn < 1946)
        cout << "Greatest Generation";</pre>
    else if (yearBorn <= 1964)
        cout << "Baby Boomer";</pre>
                                                (Demo with onlinegdb)
    }
    else if (yearBorn <= 1984)
        cout << "Generation X";</pre>
    else if (yearBorn <= 2004)
        cout << "Millennial";</pre>
    }
    else
    {
        cout << "TBD":
    return ∅;
}
```

9990

Conditionals

General format:



if (logical expression) command1; . . . else if (logical expression) command1; else command1; . . .

CSci 127 (Hunter)

3 7 May 2019 10 / 36

Sar

Logical Operators in C++

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

<ロト < 部 ト < 注 ト < 注 ト 三 三 の < ()</p>

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

<ロト < 部 ト < 注 ト < 注 ト 三 三 の < ()</p>

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

CSci 127 (Hunter)

7 May 2019 11 / 36

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

CSci 127 (Hunter)

7 May 2019 11 / 36

Today's Topics

```
//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++
- ${\hfill \circ }$ Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python
- CS Survey

Sac

イロト 不得 トイヨト イヨト 二日

In Pairs or Triples:

Predict what the following pieces of code will do:

```
//While Growth example
#include <iostream>
using namespace std;
int main ()
ł
  int population = 100;
  int year = 0;
  cout << "Year\tPopulation\n";</pre>
  while (population < 1000)
  {
      cout << year << "\t" << population << "\n";
      population = population * 2;
  }
  return 0:
CSci 127 (Hunter)
                         Lecture 13
                                                 7 May 2019
```

13 / 36

$\mathsf{C}{++} \mathsf{Demo}$

```
//While Growth example
#include <iostreams
using namespace std;
int main O
{
    int population = 100;
    int year = 0;
    cout << 'Year\tPopulation\n";
    while (population < 1000)
    {
        cout << 'Year\tPopulation < 'No";
        population = population * 2;
    }
    return 0;
}</pre>
```

(Demo with onlinegdb)

Sac

Indefinite Loops: while

```
//While Growth example
#include <iostream>
using namespace std;
int main ()
{
    int population = 100;
    int year = 0;
    cout << "Year\Population\n";
    while (population < 1000)
    {
        cout << year << "\t" << population << "\n";
        population = population * 2;
    }
    return 0;
}</pre>
```

General format:

```
while ( logical expression )
{
```

command1; command2; command3;

. . .

}

= nar

In Pairs or Triples:

Predict what the following piece of code will do:

```
//Demonstrates loops
#include <iostream>
using namespace std;
int main ()
ł
  int num;
  cout << "Enter an even number: ";</pre>
  cin >> num;
  while (num \% 2 != 0)
  {
      cout << "\nThat's odd!\n";</pre>
      cout << "Enter an even number: ";</pre>
      cin >> num;
  }
  cout << "You entered: "</pre>
        << num << ".\n";
  return 0;
}
```

$\mathsf{C}{++} \mathsf{Demo}$

```
//Demonstrates loops
#include <iostream>
using namespace std;
int main ()
  int num;
  cout << "Enter an even number: ";</pre>
  cin >> num;
  while (num \% 2 != 0)
                                               (Demo with onlinegdb)
  {
      cout << "\nThat's odd!\n":</pre>
      cout << "Enter an even number: ";</pre>
      cin >> num;
  3
  cout << "You entered: "
      << num << ".\n";
  return ∅;
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

Indefinite Loops: while

```
//Demonstrates loops
#include <iostream>
using namespace std;
int main ()
  int num;
  cout << "Enter an even number: ";</pre>
  cin >> num:
  while (num % 2 != 0)
  {
      cout << "\nThat's odd!\n";</pre>
      cout << "Enter an even number: ":
      cin >> num;
  }
  cout << "You entered: "
       << num << ".\n";
  return ∅;
3
```

General format:

```
while ( logical expression )
```

command1; command2; command3;

. . .

= nar

```
In Pairs or Triples:
```

Predict what the following pieces of code will do:

```
//Demonstrates do-while loops
#include <iostream>
using namespace std;
int main ()
{
  int num;
  do
  {
      cout << "Enter an even number: ";</pre>
      cin >> num;
  } while (num % 2 != 0);
  cout << "You entered: "
       << num << ".\n";
  return 0;
}
                                        イロト イポト イヨト イヨト
```

$\mathsf{C}{++} \mathsf{Demo}$

```
//Demonstrates do-while loops
#include <iostream>
using namespace std;
int main ()
{
  int num;
  do
  {
                                              (Demo with onlinegdb)
      cout << "Enter an even number: ";</pre>
      cin >> num;
  } while (num % 2 != 0);
  cout << "You entered: "
      << num << ".\n";
  return ∅;
}
```

= nar

<ロト <回ト < 回ト < 回ト

Indefinite Loops: do-while

```
//Demonstrates do-while loops
#include <iostream>
                                             General format:
using namespace std:
int main ()
                                             do
                                              {
  int num;
  do
                                                    command1;
  {
      cout << "Enter an even number: ";</pre>
                                                    command2;
      cin >> num:
                                                    command3;
  } while (num % 2 != 0);
                                                    . . .
  cout << "You entered: "</pre>
                                              } while ( logical expression );
       << num << ".\n";
  return 0:
}
```

= nar

Today's Topics

```
//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++
- ${\scriptstyle \bullet }$ Indefinite Loops in C++
- Recap: C++ & Python
- CS Survey

Sac

イロト 不得 トイヨト イヨト 二日

• I/O:

```
//Arother (-+, program, Demostrates loops

with monospore txd;

in team ()

{ mit j;

for (1 = 0; i < 4; i++)

{ cout << "The world turned upside down...\n";

for (1 = 0; i < 4; j > 0; j --)

{ cout << j <= ";

cout << "Rlast off|] << end;

} return 0;

}
```

500

• I/O: cin >> ...;

▲□▶ ▲□▶ ▲ヨ▶ ▲ヨ▶ ヨ - のの⊙

• I/O: cin >> ...; & cout << ...;

200

• I/O: cin >> ...; & cout << ...;

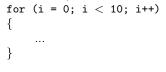
• Definite loops:

200

イロト 不得 トイヨト イヨト 二日

• I/O: cin >> ...; & cout << ...;

• Definite loops:



//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;

```
 \begin{array}{l} (\text{in tani } C) \\ & (\text{in } i,j) \\ & (\text{in } i,j) \\ & (\text{in } 0; i = 0; i < 4; i \leftrightarrow ) \\ & (\text{in } 0; i = 0; i < 4; i \leftrightarrow ) \\ & (\text{out } < \cdots \text{ in } world turned upside down,... \end{in } vin'; \\ & (\text{out } < i 18; j > 0; j \sim ) \\ & (\text{out } < j < < " "; \\ & (\text{cout } < "Blast off ||" << enl; \\ & (\text{return } 0; \\ \end{in } vint \en
```

= nar

```
• I/O: cin >> ...; & cout << ...;
Definite loops:
  for (i = 0; i < 10; i++)
  í
       ...
  }
Conditionals:
```

//Another C++ program; Demonstrates loops #include <iostream> using namespace std;

```
int main ()
 int i,j;
 for (i = 0; i < 4; i + )
      cout << "The world turned upside down...\n";</pre>
 3
  for (j = 10; j > 0; j - -)
 ł
     cout << j << " ";
 cout << "Blast off!!" << endl:
 return 0;
```

Sac

```
#include <iostream>
using namespace std;
int main ()
 int i,j;
 for (i = 0; i < 4; i + )
 {
      cout << "The world turned upside down...\n";</pre>
  for (j = 10; j > 0; j - -)
     cout << j << " ";
 cout << "Blast off!!" << endl:
 return 0;
```

//Another C++ program: Demonstrates loops

```
I/O: cin >> ...; & cout << ...;
</pre>
Definite loops:
  for (i = 0; i < 10; i++)
       ...
Conditionals:
  if (logical expression)
  ł
  else
```

Sac

```
//Another C++ program: Demonstrates loops
#include <iostream>
using namespace std;
int main ()
 int i,j;
 for (i = 0; i < 4; i++)
 {
      cout << "The world turned upside down...\n";</pre>
  for (j = 10; j > 0; j - -)
     cout << j << " ":
 cout << "Blast off!!" << endl:
 return 0;
```

```
• I/O: cin >> ...; & cout << ...;</pre>
Definite loops:
  for (i = 0; i < 10; i++)
       ...
Conditionals:
  if (logical expression)
  ſ
  else
Indefinite loops:
```

Sac

CSci 127 (Hunter)

```
• I/O: cin >> ...; & cout << ...;</pre>
Definite loops:
  for (i = 0; i < 10; i++)
        ...
Conditionals:
  if (logical expression)
  else
Indefinite loops:
  while (logical expression)
        ...
                      イロト イポト イヨト イヨト
                                             3
                                                Sac
     Lecture 13
                                     7 May 2019
                                                23 / 36
```

• Rewrite this program in C++:

```
for i in range(2017, 2000, -2):
    print("Year is", i)
```

• Rewrite this program in Python:

```
#include <iostream>
using namespace std;
int main()
  for (int i = 1; i < 50; i++)
   cout << i << endl;
  }
  return 0;
```

CSci 127 (Hunter)

3 7 May 2019 24 / 36

4 E b

• *Rewrite this program in C++:*

for i in range(2017, 2000, -2):
 print("Year is", i)

• *Rewrite this program in C++:*

for i in range(2017, 2000, -2):
 print("Year is", i)

#include <iostream>
using namespace std;

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

• *Rewrite this program in C++:*

```
for i in range(2017, 2000, -2):
    print("Year is", i)
```

```
#include <iostream>
using namespace std;
int main()
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

• Rewrite this program in C++:

```
for i in range(2017, 2000, -2):
    print("Year is", i)
```

```
#include <iostream>
using namespace std;
int main()
{
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

```
• Rewrite this program in C++:
```

```
for i in range(2017, 2000, -2):
    print("Year is", i)
```

```
#include <iostream>
using namespace std;
int main()
{
  for (int i = 2017; i >= 2000; i=i-2)
```

E ∽QQ

- - E - b

< □ > < 同 > < 三 >

```
• Rewrite this program in C++:
```

```
for i in range(2017, 2000, -2):
    print("Year is", i)
```

```
#include <iostream>
using namespace std;
int main()
{
   for (int i = 2017; i >= 2000; i=i-2)
    {
      cout << "Year is" << i << endl;
   }
}</pre>
```

```
• Rewrite this program in C++:
```

```
for i in range(2017, 2000, -2):
    print("Year is", i)
```

```
#include <iostream>
using namespace std;
int main()
{
   for (int i = 2017; i >= 2000; i=i-2)
     {
        cout << "Year is" << i << endl;
     }
     return 0;
}</pre>
```

CSci 127 (Hunter)

> < ≣ > ≣ ∽ < < 7 May 2019 25 / 36

```
• Rewrite this program in Python:
```

```
#include <iostream>
using namespace std;
int main()
{
   for (int i = 1; i < 50; i++)
    {
      cout << i << endl;
   }
   return 0;
}</pre>
```

```
• Rewrite this program in Python:
```

```
#include <iostream>
using namespace std;
int main()
  for (int i = 1; i < 50; i++)
   cout << i << endl;
  }
 return 0;
for i in range(1, 50):
```

Sar

```
• Rewrite this program in Python:
```

```
#include <iostream>
using namespace std;
int main()
  for (int i = 1; i < 50; i++)
   cout << i << endl;
  }
 return 0;
for i in range(1, 50):
    print(i)
```

CSci 127 (Hunter)

7 May 2019 26 / 36

Sac

```
• Python: what is the output?
year = 2016
if year % 4 == 0 and \
      (not (year % 100 == 0) or (year % 400 == 0)):
      print("Leap!!")
print("Year")
```

 Write a C++ program that asks the user the number of times they plan to ride transit this week. Your program should then print if it is cheaper to buy single ride metro cards or 7-day unlimited card. (The 7-day card is \$31.00, and the cost of single ride, with bonus, is \$2.48).

CSci 127 (Hunter)

Lecture 13

7 May 2019 27 / 36

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙

```
• Python: what is the output?
year = 2016
if year % 4 == 0 and \
        (not (year % 100 == 0) or (year % 400 == 0)):
        print("Leap!!")
print("Year")
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

```
• Python: what is the output?
year = 2016
if year % 4 == 0 and \
        (not (year % 100 == 0) or (year % 400 == 0)):
        print("Leap!!")
print("Year")
```

Working out the arithmetic and logic:

```
• Python: what is the output?
year = 2016
if year % 4 == 0 and \
        (not (year % 100 == 0) or (year % 400 == 0)):
        print("Leap!!")
print("Year")
```

Working out the arithmetic and logic:

```
• year % 4 is 504.
```

```
• Python: what is the output?
year = 2016
if year % 4 == 0 and \
      (not (year % 100 == 0) or (year % 400 == 0)):
      print("Leap!!")
print("Year")
```

Working out the arithmetic and logic:

• year % 4 is 504.

• 504 \neq 0 so the first part of the and is False.

```
• Python: what is the output?
year = 2016
if year % 4 == 0 and \
      (not (year % 100 == 0) or (year % 400 == 0)):
      print("Leap!!")
print("Year")
```

Working out the arithmetic and logic:

- year % 4 *is* 504.
- 504 \neq 0 so the first part of the and is False.
- Since (False and anything) is False, the expression is False.

CSci 127 (Hunter)

7 May 2019 28 / 36

```
• Python: what is the output?
year = 2016
if year % 4 == 0 and \
      (not (year % 100 == 0) or (year % 400 == 0)):
      print("Leap!!")
print("Year")
```

Working out the arithmetic and logic:

- year % 4 *is* 504.
- 504 \neq 0 so the first part of the and is False.
- Since (False and anything) is False, the expression is False. (There's no need to figure out the rest of the expression.)

CSci 127 (Hunter)

7 May 2019 28 / 36

イロト 不得 トイヨト イヨト ヨー のくや

```
• Python: what is the output?
year = 2016
if year % 4 == 0 and \
      (not (year % 100 == 0) or (year % 400 == 0)):
      print("Leap!!")
print("Year")
```

Working out the arithmetic and logic:

- year % 4 is 504.
- 504 \neq 0 so the first part of the and is False.
- Since (False and anything) is False, the expression is False. (There's no need to figure out the rest of the expression.)
- Never enter the if-clause and go to the next line.

CSci 127 (Hunter)

7 May 2019 28 / 36

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙

```
• Python: what is the output?
year = 2016
if year % 4 == 0 and \
      (not (year % 100 == 0) or (year % 400 == 0)):
      print("Leap!!")
print("Year")
```

Working out the arithmetic and logic:

- year % 4 is 504.
- 504 \neq 0 so the first part of the and is False.
- Since (False and anything) is False, the expression is False. (There's no need to figure out the rest of the expression.)
- Never enter the if-clause and go to the next line.
- Only thing printed is: Year

CSci 127 (Hunter)

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

 Your program should then print if it is cheaper to buy single ride metro cards (\$2.48 per ride) or 7-day unlimited card (\$31.00).
 #include <iostream> using namespace std;

Sac

 Your program should then print if it is cheaper to buy single ride metro cards (\$2.48 per ride) or 7-day unlimited card (\$31.00).
 #include <iostream> using namespace std; int main()

Sac

 Your program should then print if it is cheaper to buy single ride metro cards (\$2.48 per ride) or 7-day unlimited card (\$31.00).
 #include <iostream> using namespace std; int main() {

```
int rides;
```

Sac

• Your program should then print if it is cheaper to buy single ride metro cards
 (\$2.48 per ride) or 7-day unlimited card (\$31.00).
 #include <iostream>
 using namespace std;
 int main()
 {
 int rides;
 cout << "Enter number of rides:";</pre>

Sac

• Your program should then print if it is cheaper to buy single ride metro cards
 (\$2.48 per ride) or 7-day unlimited card (\$31.00).
 #include <iostream>
 using namespace std;
 int main()
 {
 int rides;
 cout << "Enter number of rides:";
 cin >> rides;

Sac

• Your program should then print if it is cheaper to buy single ride metro cards
(\$2.48 per ride) or 7-day unlimited card (\$31.00).
#include <iostream>
using namespace std;
int main()
{
 int rides;
 cout << "Enter number of rides:";
 cin >> rides;
 if (2.48 * rides < 31.00)</pre>

200

• Your program should then print if it is cheaper to buy single ride metro cards (\$2.48 per ride) or 7-day unlimited card (\$31.00). #include <iostream> using namespace std; int main() int rides; cout << "Enter number of rides:";</pre> cin >> rides;if (2.48 * rides < 31.00)cout << "Cheaper to buy single ride metro cards.\n";

• Your program should then print if it is cheaper to buy single ride metro cards (\$2.48 per ride) or 7-day unlimited card (\$31.00). #include <iostream> using namespace std; int main() int rides; cout << "Enter number of rides:";</pre> cin >> rides;if (2.48 * rides < 31.00)cout << "Cheaper to buy single ride metro cards.\n"; } else

• Your program should then print if it is cheaper to buy single ride metro cards (\$2.48 per ride) or 7-day unlimited card (\$31.00). #include <iostream> using namespace std; int main() int rides; cout << "Enter number of rides:";</pre> cin >> rides;if (2.48 * rides < 31.00)cout << "Cheaper to buy single ride metro cards.\n"; else cout << "Cheaper to buy 7-day unlimited card.\n";

• Your program should then print if it is cheaper to buy single ride metro cards (\$2.48 per ride) or 7-day unlimited card (\$31.00). #include <iostream> using namespace std; int main() int rides; cout << "Enter number of rides:";</pre> cin >> rides;if (2.48 * rides < 31.00)cout << "Cheaper to buy single ride metro cards.\n"; else cout << "Cheaper to buy 7-day unlimited card.\n"; return 0; イロト イポト イヨト イヨト = nac CSci 127 (Hunter) Lecture 13 7 May 2019 29 / 36

• Write Python code that repeatedly prompts for a non-empty string.

• Write C++ code that repeatedly prompts until an odd number is entered.

CSci 127 (Hunter)

Lecture 13

7 May 2019 30 / 36

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > <

• Write Python code that repeatedly prompts for a non-empty string.

Sac

• Write Python code that repeatedly prompts for a non-empty string.

Sac

<ロト < 回 > < 回 > < 回 > < 回 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

• Write Python code that repeatedly prompts for a non-empty string.

```
s = ""
while s == "":
```

• Write Python code that repeatedly prompts for a non-empty string.

```
s = ""
while s == "":
    s = input("Enter a non-empty string: ")
```

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

• Write Python code that repeatedly prompts for a non-empty string.

```
s = ""
while s == "":
    s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

- Write Python code that repeatedly prompts for a non-empty string. s = "" while s == "": s = input("Enter a non-empty string: ") print("You entered: ", s)
- Write C++ code that repeatedly prompts until an odd number is entered.

• Write Python code that repeatedly prompts for a non-empty string. s = "" while s == "": s = input("Enter a non-empty string: ")

```
print("You entered: ", s)
```

 Write C++ code that repeatedly prompts until an odd number is entered.
 #include <iostream> using namespace std;

Write Python code that repeatedly prompts for a non-empty string.
 s = ""
 while s == "":

```
s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

 Write C++ code that repeatedly prompts until an odd number is entered.
 #include <iostream> using namespace std; int main()

Write Python code that repeatedly prompts for a non-empty string.
 s = ""
 while s == "":

```
s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

• Write C++ code that repeatedly prompts until an odd number is entered.
#include <iostream>
using namespace std;
int main()
{
 int num = 0;

◆□▶ ◆□▶ ◆三▶ ◆三▶ ─ 三 ● ○○○

Write Python code that repeatedly prompts for a non-empty string.
 s = ""
 while s == "":

```
s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

• Write C++ code that repeatedly prompts until an odd number is entered.
#include <iostream>
using namespace std;
int main()
{
 int num = 0;
 while (num % 2 == 0)

◆□▶ ◆□▶ ◆三▶ ◆三▶ ─ 三 ● ○○○

Write Python code that repeatedly prompts for a non-empty string.
 s = ""

```
while s == "":
   s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

• Write C++ code that repeatedly prompts until an odd number is entered.
#include <iostream>
using namespace std;
int main()
{
 int num = 0;
 while (num % 2 == 0)
 {
}

```
cout << "Enter an odd number:";</pre>
```

CSci 127 (Hunter)

7 May 2019 31 / 36

◆□▶ ◆□▶ ◆三▶ ◆三▶ ─ 三 ● ○○○

Write Python code that repeatedly prompts for a non-empty string.
 s = ""

```
while s == "":
   s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

• Write C++ code that repeatedly prompts until an odd number is entered.
#include <iostream>
using namespace std;
int main()
{
 int num = 0;
 while (num % 2 == 0)
 {
 cout << "Enter an odd number:";</pre>

```
cin >> num;
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ ─ 三 ● ○○○

7 May 2019

31 / 36

Write Python code that repeatedly prompts for a non-empty string.
 s = ""

```
while s == "":
   s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

• Write C++ code that repeatedly prompts until an odd number is entered.
#include <iostream>
using namespace std;
int main()
{
 int num = 0;
 while (num % 2 == 0)
 {
 cout << "Enter an odd number:";
 cin >> num;

```
CSci 127 (Hunter)
```

= nac

Write Python code that repeatedly prompts for a non-empty string.
 s = ""
 while s == "":

```
s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

• Write C++ code that repeatedly prompts until an odd number is entered. #include <iostream> using namespace std; int main() int num = 0: while (num % 2 == 0)cout << "Enter an odd number:";</pre> cin >> num;return 0; イロト イポト イヨト イヨト E ∽QQ CSci 127 (Hunter) Lecture 13 7 May 2019 31 / 36

Today's Topics

```
//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++
- Recap: C++ & Python
- CS Survey

Sac

Collin Waldoch



(Image from New Yorker)

CSci 127 (Hunter)

Lecture 13

3 7 May 2019 33 / 36

990

Collin Waldoch



(Image from New Yorker)

 Brief overview of Citi Bike & Bike Angels

996

Collin Waldoch



(Image from New Yorker)

- Brief overview of Citi Bike & Bike Angels
- What Collin does and loves about Bike Angels.

Sac

Collin Waldoch



(Image from New Yorker)

- Brief overview of Citi Bike & Bike Angels
- What Collin does and loves about Bike Angels.
- Design challenge: work in pairs or triples with Bike Angels & UTAs.

イロト イロト イヨト イ

Collin Waldoch



(Image from New Yorker)

- Brief overview of Citi Bike & Bike Angels
- What Collin does and loves about Bike Angels.
- Design challenge: work in pairs or triples with Bike Angels & UTAs.
- 12:30pm: Informal Q&A in 631 Hunter East (inside library).

Мар



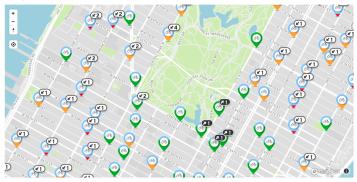
Э 7 May 2019 34 / 36

 $\exists \rightarrow$

900

4 日 × 4 日 × 4 日 ×

Мар



• Design an algorithm to find mostly full stations.

CSci 127 (Hunter)

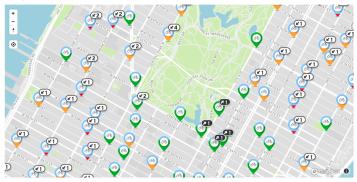
Lecture 13

7 May 2019 34 / 36

1

Image: A math a math

Мар



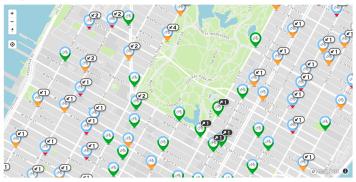
- Design an algorithm to find mostly full stations.
- Design an algorithm to maximize points earned.

CSci 127 (Hunter)

Lecture 13

. < □ > . < □</p>

Мар



- Design an algorithm to find mostly full stations.
- Design an algorithm to maximize points earned.
- Note: map and photo form on back of lecture slip.

CSci 127 (Hunter)

. (□) < □</p>



• Lightning rounds:

CSci 127 (Hunter)

Lecture 13

7 May 2019 35 / 36

3

590



- Lightning rounds:
 - write as much you can for 60 seconds;

-

200

A D > A D > A D >
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A



- Lightning rounds:
 - write as much you can for 60 seconds;
 - followed by answer; and

CSci 127 (Hunter)

Lecture 13

7 May 2019 35 / 36

-

イロト イロト イヨト イ



- Lightning rounds:
 - write as much you can for 60 seconds;
 - ► followed by answer; and
 - ► repeat.



- Lightning rounds:
 - write as much you can for 60 seconds;
 - followed by answer; and
 - ▶ repeat.
- Past exams are on the webpage (under Final Exam Information).

CSci 127 (Hunter)

Lecture 13

7 May 2019 35 / 36

イロト イロト イヨト イ



- Lightning rounds:
 - write as much you can for 60 seconds;
 - followed by answer; and
 - ► repeat.
- Past exams are on the webpage (under Final Exam Information).
- We'll start with: Fall 17, Version 3.

CSci 127 (Hunter)

イロト イロト イヨト イ

Writing Boards



• Return writing boards as you leave...

CSci 127 (Hunter)

7 May 2019 36 / 36