CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

イロト イポト イヨト イヨト

CSci 127 (Hunter)

Lecture 13

5 December 2018 1 / 41

3

990

Today's Topics

//Another C++ program, demonstrating I/O & arithmetic Minclude <iostreamo using nomespace std; int main Q

```
{ flost kg, lbs; cout << "Enter kg: "; cin >> kg; "; 
 bs = 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++

• Review: Design & Final Questions

```
//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;
3
```

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~

• 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
wining ammespace std;
int main O
{ float kg, lbs;
coat << "Enter kg: ";
tbs kg * 2.2;
coat << "enter kg: " << lbs << "\n\n";
} eturn 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: ";
  chip: kg; 2.2;
  coat << endl << "Lbs: " << lbs << "\n\n";
  returm 0;
```

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

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cins - kg; 2.2;
  coat << endl << "lbs: " << lbs << "\n\n";
  returm 0;
```

Sac

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: ";
  cins - kg; 2.2;
  coat << endl << "lbs: " << lbs << "\n\n";
  returm 0;
```

Sac

- 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: ";
  cins - kg; 2.2;
  coat << endl << "lbs: " << lbs << "\n\n";
  returm 0;
```

Sac

<pre>//Another C++ program, demonstrating I/O & arithmeti #include <iostream> using namespace std;</iostream></pre>
int main ()
float ka 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:

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

<pre>/Another C++ program, demonstrating I/O & arithmetic include <iostream> sing namespace std;</iostream></pre>
nt main ()
float ka 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!!";

E Sac

<pre>//Another C++ program, demonstrating I/O & arithmet #include <iostream> using namespace std;</iostream></pre>
<pre>int main () { fort kg, lbs; cout < "Enter kg: "; cin >> kg; lbs = kg * 2.2; cout << endl << "Lbs; " << lbs << "\n\n"; return 0; }</pre>
1

- 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:

≡ ∽ar

<pre>//Another C++ program, demonstrating I/O & arithmet #include <iostream> using namespace std;</iostream></pre>
int main () {
float kg, lbs;
cout << enter kg: ; cin >> kg;
lbs = kg * 2.2;
cout << endl << "Lbs: " << lbs << "\n\n";
recurr v,

- 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;

3

200

<pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre>
$\label{eq:constraint} \begin{array}{l} \text{int main } O \\ \{ \text{fott } k_0, \text{ bs}; \\ \text{cont} < < \mbox{``Enter } k_0; \ `; \\ \mbox{cin } s > k_0; \\ \mbox{bs} < k_0 \neq 2.2; \\ \mbox{cont} < \mbox{end} < < \mbox{``Lbs}: \ ` < \mbox{lbs} < < \ ``urin"; \\ \mbox{return } 0; \\ \mbox{return } 0; \\ \end{array} \end{array}$

- 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:

3

Sac

<pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre>
int main () { float ka lhr:
cout << "Enter 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;

<pre>//Another L++ program, demonstrating I/U & arithme #include <iostream> using namespace std;</iostream></pre>
<pre>int main () { for the lag, lbs; cost << "Enter kg; "; for >< "Enter kg; "; lbs = kg * 2.2; cost << 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 1/0 & arithme #include <iostream> using namespace std;</iostream></pre>
<pre>int main () { force 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: 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 ka, lbs:
cout << "Enter kg: "; cin >> kg:
lbs = kg * 2.2;
cout << endl << "Lps: " << 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++) {...}</pre>
- Blocks of code uses '{' and '}'.

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~

<pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre>
int main () {
rloat kg, LDS;
cin >> ka:
lbs = kg * 2.2;
cout co endl co "libs: " co libs co "\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 Winclude <iostreamo using nomespace std; int main Q

```
{

flost 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++

• Review: Design & Final Questions

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 \leq 2004)
        cout << "Millennial":</pre>
    }
    else
        cout << "TBD":
```

```
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

return 0: CSci 127 (Hunter)

Lecture 13

5 December 2018 6 / 41

3

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";
    }
    else if (yearBorn <= 1984)
         cout << "Generation X";</pre>
    else if (yearBorn <= 2004)
         cout << "Millennial";</pre>
    }
    else
    {
         cout << "TBD":
    return ∅;
}
```

(Demo with onlinegdb)

3 Sac

Conditionals

General format:



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

CSci 127 (Hunter)

5 December 2018 8 / 41

3

Logical Operators in C++

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

<ロト < 同ト < 臣ト < 臣ト = 臣 = のへで

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)

<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

not (!)

in1		returns:	
!	False	True	
!	True	False	

CSci 127 (Hunter)

5 December 2018 9 / 41

Today's Topics

//Another C++ program, demonstrating I/O & arithmetic Minclude <iostreamo using nomespace std; int main Q

```
 \begin{cases} & f \mbox{ord} x_{1} \mbox{ bs; } \\ \mbox{cont} < \mbox{ "Enter kg: "; } \\ \mbox{cin >> kg: } \\ \mbox{lbs} = kg \mbox{ * 2.2; } \\ \mbox{cont} < \mbox{endl sc "Lbs: " << lbs << "\n\n"; } \\ \mbox{return 0; } \end{cases}
```

• Recap: I/O & Definite Loops in C++

• Conditionals in C++

• Indefinite Loops in C++

• Review: Design & Final Questions

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
                                              5 December 2018
```

11 / 41

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

```
//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)

イロト 不良 トイヨト イヨト ヨー のくや

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 ()
{
 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: "
       << num << ".\n";
  return ∅;
```

CSci 127 (Hunter)

= nar

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

```
//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: "</pre>
      << num << ".\n";
  return ∅;
```

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

Indefinite Loops: while

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

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 ()
  ł
   int num;
    do
    {
        cout << "Enter an even number: ";</pre>
        cin >> num;
    } while (num % 2 != 0);
    cout << "You entered: "</pre>
         << num << ".\n";
    return 0;
  }
```

CSci 127 (Hunter)

5 December 2018 17 / 41

≡ ∽ar

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

```
//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
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 Minclude <iostreamo using nomespace std; int main Q

```
 \begin{cases} \label{eq:control of the set of the set
```

• Recap: I/O & Definite Loops in C++

• Conditionals in C++

• Indefinite Loops in C++

• Review: Design & Final Questions

• I/O:

• I/O: cin >> ...;

• I/O: cin >> ...; & cout << ...;

• I/O: cin >> ...; & cout << ...;

• Definite loops:

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

#iclude -(otreteme)

with memopole tid;

in teal O

( int i,j;

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

f out << "The world turned upside down...\n";

for (j = 10; j > 0; j--)

f out << "The world turned upside down...\n";

for (j = 10; j > 0; j--)

f out << "Illast off|] << endl;

return 0;
```

• I/O: cin >> ...; & cout << ...;

• Definite loops:



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

```
 \begin{array}{l} (\text{in that } Q) \\ \left\{ \begin{array}{l} (\text{in } i,j) \\ (\text{in } i,j) \\ (\text{in } i,i \in i; i < 4; i \leftrightarrow ) \\ \left\{ \begin{array}{l} (\text{out } \ll \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \ll \text{'The world turned upside down,...\n''; } \\ \left\{ \begin{array}{l} \text{out } \ll \text{'T} \\ (\text{out } \ll \text{'T} \in \mathbb{R}^{n}) \\ \text{out } \ll \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \ll \text{'The world turned upside down,...\n''; } \\ \text{out } \ll \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \ll \text{'The world turned upside down,...\n''; } \\ \text{out } \ll \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \ll \text{'The world turned upside down,...\n''; } \\ \text{out } \ll \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \ll \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \ll \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \ll \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \bowtie \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \bowtie \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \bowtie \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \bowtie \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \bowtie \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \bowtie \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \bowtie \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \u \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \u \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \u \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \u \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \u \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \u \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \u \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \u \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \u \text{'The world turned upside down,...\n''; } \end{array} \right. \\ \left\{ \begin{array}{l} \text{out } \u \text{'The world turned up
```

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

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

```
 \begin{array}{l} (\text{in tank } Q) \\ \left\{ \begin{array}{l} \text{int } \{i,j\} \\ \text{for } (i=0;\ i<4;\ i+*) \\ f \\ \\ \text{cout } \ll \text{ The world turned upside down...\vn';} \\ \end{array} \right. \\ \left\{ \begin{array}{l} \text{for } (j=10;\ j>0;\ j-\cdot) \\ \\ \\ \text{cout } \ll \text{ These off} \\ \text{cout } \ll \text{ These off} \\ \end{array} \right. \\ \left\{ \begin{array}{l} \text{cout } \ll \text{ i} \\ \text{cout } \ll \text{ These off} \\ \text{cout } \ll \text{ rel} \\ \end{array} \right. \\ \end{array} \right.
```

//Another C++ program; Demonstrates loops

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

= nar

```
\label{eq:constraints} C = constraints = coops \\ which der closer constraints = coops \\ which der closer constraints = coops \\ int nain (D) \\ \{ int (j; 0; i < 4; i++) \\ for (j = 0; i < 4; i++) \\ for (j = 0; i < 0; i <
```

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

= nar

```
I/O: cin >> ...; & cout << ...;
</pre>
                                           Definite loops:
                                               for (i = 0; i < 10; i++)
                                                       ...
                                           Conditionals:
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;
                                               if (logical expression)
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 << " ":
                                               else
 cout << "Blast off!!" << endl:
 return 0;
                                           Indefinite loops:
                                               while (logical expression)
                                                       ...
                                                                            イロト 不得下 イヨト イヨト
                                                                                                              3
                                                                                                                 200
    CSci 127 (Hunter)
                                                    Lecture 13
                                                                                            5 December 2018
                                                                                                                  21 / 41
```

Design Question: Street Trees



Design an algorithm that prints out all the street trees in your zip code (using NYC Urban Forest OpenData).

CSci 127 (Hunter)

Lecture 13

5 December 2018 22 / 41

Sac

Design Question: Street Trees

1	created_at	tree_id	block_id	the_geor	n c	urb_loc	status	health	spc_latin	spc_common	address	zipcode	zip_city	cb_num	borococ boroname	nta	nta_name	boro_ct	state	Latitude	longitude
2	8/27/15	180683	348711	POINT (-	73.8 C	InCurb	Alive	Fair	Acer rubrum	red maple	108-005 70 AVENUE	11375	Forest Hills	406	4 Queens	QN17	Forest Hills	4073900	New York	40.7230918	-73.844215
3	9/3/15	200540	315986	POINT (-	73.8 C	InCurb	Alive	Fair	Quercus pals	pin oak	147-074 7 AVENUE	11357	Whitestone	407	4 Queens	QN49	Whitestone	4097300	New York	40.7941107	-73.818679
4	9/5/15	204026	218365	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	390 MORGAN AVEN	11211	Brooklyn	301	3 Brooklyn	BK90	East William	3044900	New York	40.7175807	-73.936608
5	9/5/15	204337	217969	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	1027 GRAND STREE	11211	Brooklyn	301	3 Brooklyn	BK90	East William	3044900	New York	40.7135375	-73.934456
6	8/30/15	189565	223043	POINT (-	73.9 C	InCurb	Alive	Good	Tilia america	American lin	603 6 STREET	11215	Brooklyn	306	3 Brooklyn	BK37	Park Slope-C	3016500	New York	40.6667778	-73.975979
7	8/30/15	190422	106099	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	8 COLUMBUS AVEN	10023	New York	107	1 Manhattan	MN14	Lincoln Squa	1014500	New York	40.7700456	-73.98495
8	8/30/15	190426	106099	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	120 WEST 60 STREE	10023	New York	107	1 Manhattan	MN14	Lincoln Squa	1014500	New York	40.7702097	-73.985338
9	9/7/15	208649	103940	POINT (-	73.9 C	InCurb	Alive	Good	Tilia america	American lin	311 WEST 50 STREE	10019	New York	104	1 Manhattan	MN15	Clinton	1012700	New York	40.7627239	-73.987297
10	9/8/15	209610	407443	POINT (-	74.0 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	65 JEROME AVENUI	10305	Staten Island	502	5 Staten Islan	d SI 14	Grasmere-A	r 5006400	New York	40.5965793	-74.076255
11	8/31/15	192755	207508	POINT (-	73.9 C	OffsetFromC	Alive	Fair	Platanus x ac	London plan	638 AVENUE Z	11223	Brooklyn	313	3 Brooklyn	BK26	Gravesend	3037402	New York	40.5863573	-73.969744
12	9/5/15	203719	302371	POINT (-	73.9 C	InCurb	Alive	Good	Platanus x ad	London plan	20-025 24 STREET	11105	Astoria	401	4 Queens	QN72	Steinway	4010500	New York	40.7824282	-73.911171
13	9/5/15	203726	302371	POINT (*	73.9 C	InCurb	Alive	Poor	Platanus x ad	London plan	20-055 24 STREET	11105	Astoria	401	4 Queens	QN72	Steinway	4010500	New York	40.7817351	-73.91202
14	9/1/15	195202	415896	POINT (-	74.1 C	InCurb	Alive	Fair	Platanus x ad	London plan	35 FENWAY CIRCLE	10308	Staten Island	503	5 Staten Islan	d \$154	Great Kills	5014607	New York	40.5571026	-74.16267
15	8/30/15	189465	219493	POINT (-	73.9 C	InCurb	Alive	Good	Platanus x ac	London plan	100 WAVERLY AVEN	11205	Brooklyn	302	3 Brooklyn	BK69	Clinton Hill	3019100	New York	40.6947331	-73.968211
16	8/31/15	192998	211160	POINT (-	73.9 C	InCurb	Alive	Fair	Platanus x ad	London plan	2126 UNION STREET	11212	Brooklyn	316	i 3 Brooklyn	BK81	Brownsville	3090000	New York	40.6643174	-73.92113
17	8/30/15	189834	219505	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	449 MYRTLE AVENU	11205	Brooklyn	302	3 Brooklyn	BK69	Clinton Hill	3019100	New York	40.6933142	-73.967601
18	9/5/15	204208	204913	POINT (-	73.9 C	InCurb	Alive	Fair	Ginkgo bilob	ginkgo	8797 25 AVENUE	11214	Brooklyn	311	3 Brooklyn	BK29	Bensonhurst	3030400	New York	40.5937876	-73.991597
19	8/20/15	161339	230864	POINT (-	73.9 C	nCurb	Alive	Good	Gleditsia tria	honeylocust	1601 CHURCH AVEN	11226	Brooklyn	314	3 Brooklyn	BK42	Flatbush	3051200	New York	40.6487877	-73.964524
20	8/29/15	187311	342816	POINT (-	73.8 C	InCurb	Alive	Fair	Gleditsia tria	honeylocust	55-026 96 STREET	11373	Elmhurst	404	4 Queens	QN25	Corona	4044302	New York	40.7376462	-73.8653
21	9/7/15	208201	219258	POINT (*	73.9 C	InCurb	Alive	Good	Ginkgo bilob	ginkgo	206 CARLTON AVEN	11205	Brooklyn	302	3 Brooklyn	BK68	Fort Greene	3018300	New York	40.6914992	-73.972588
22	8/31/15	193093	225564	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	367 PROSPECT AVE	11215	Brooklyn	307	3 Brooklyn	BK37	Park Slope-C	3014900	New York	40.6612388	-73.985889
23	8/31/15	193310	107600	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	170 EAST 75 STREET	10021	New York	108	1 Manhattan	MN40	Upper East S	1012800	New York	40.7721715	-73.960456
24	9/3/15	199760	107038	POINT (-	73.9 C	InCurb	Alive	Fair	Quercus phe	willow oak	2163 BROADWAY	10024	New York	107	1 Manhattan	MN12	Upper West	1016300	New York	40.7820872	-73.980964
25	9/5/15	204325	415427	POINT (-	74.1 C	InCurb	Alive	Good	Acer pseudo	sycamore ma	301 THOMAS STREE	10306	Staten Island	503	5 Staten Islan	d \$125	Oakwood-O	5013800	New York	40.5688212	+74.138563
26	8/30/15	189700	223525	POINT (-	73.9 C	InCurb	Alive	Good	Quercus phe	willow oak	26 8 AVENUE	11217	Brooklyn	306	i 3 Brooklyn	BK37	Park Slope-G	3015900	New York	40.6748392	-73.972184
27	9/5/15	205044	206800	POINT (-	73.9 C	InCurb	Alive	Good	Acer ginnala	Amur maple	134 BAY 40 STREET	11214	Brooklyn	311	3 Brooklyn	BK29	Bensonhurst	3030200	New York	40.5949363	-73.989171
28	8/31/15	192942	310298	POINT (-	73.8 C	InCurb	Alive	Good	Platanus x ad	London plan	84-036 127 STREET	11415	Kew Garden	409	4 Queens	Q/N60	Kew Garden	4013800	New York	40.7065335	-73.824992
29	9/1/15	195265	227489	POINT (-	73.9 C	InCurb	Alive	Good	Platanus x ad	London plan	401 AVENUE O	11230	Brooklyn	312	3 Brooklyn	BK46	Ocean Parks	3044000	New York	40.6119046	-73.970427
30	9/4/15	201555	102919	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	71 STANTON STREE	10002	New York	103	1 Manhattan	MN27	Chinatown	1003601	New York	40.7218074	-73.98983
31	9/7/15	208407	345512	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	1817 DE KALB AVEN	11385	Ridgewood	405	4 Queens	QN20	Ridgewood	4053900	New York	40.7080401	-73.915497
32	8/30/15	189164	223133	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	456 5 AVENUE	11215	Brooklyn	306	3 Brooklyn	BK37	Park Slope C	3013900	New York	40.6688265	-73.986703
33	8/29/15	184375	505172	POINT (*	73.8 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	2022 LA FONTAINE	10457	Bronx	206	2 Bronx	BX17	East Tremor	2037504	New York	40.8479471	-73.893382
34	8/29/15	187056	106107	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	1880 BROADWAY	10023	New York	107	1 Manhattan	MN14	Lincoln Squa	1014900	New York	40.7703961	-73.981627
35	8/29/15	187058	106107	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	1 WEST 62 STREET	10023	New York	107	1 Manhattan	MN14	Lincoln Squa	1014900	New York	40.7702272	-73.981218
36	8/26/15	178659	406774	POINT (-	74.0 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	434 RARITAN AVEN	10305	Staten Island	502	5 Staten Islan	8 5136	Old Town-D	5007000	New York	40.5881073	-74.086678
37	8/30/15	190423	106099	POINT (-	73.9 C	InCurb	Alive	Good	Gleditsia tria	honeylocust	120 WEST 60 STREE	10023	New York	107	1 Manhattan	MN14	Lincoln Squa	1014500	New York	40.770087	-73.985048
38	8/30/15	190425	106099	POINT (-	73.9 C	Curb	Alive	Good	Gleditsia tria	honeylocust	120 WEST 60 STREE	10023	New York	107	1 Manhattan	MN14	Lincoln Squa	1014500	New York	40.7701806	-73.985269
39	9/5/15	204202	415447	POINT (-	74.1 C	InCurb	Alive	Fair	Platanus x ad	London plan	328 THOMAS STREE	10306	Staten Island	503	5 Staten Islan	d \$125	Oakwood-O	5013800	New York	40.5688252	-74.139062
40	9/4/15	201494	406253	POINT (-	74.1 C	InCurb	Alive	Good	Platanus x ad	London plan	6 ALLISON PLACE	10306	Staten Island	502	5 Staten Islan	d \$145	New Dorp-N	5012200	New York	40.5780741	-74.114746
41	9/7/15	208346	106403	POINT (-	73.9 C	offsetFromC	Alive	Good	Platanus x ad	London plan	418 COLUMBUS AV	10024	New York	107	1 Manhattan	MN12	Upper West	1016500	New York	40.7825867	-73.97484

Design an algorithm that prints out all the street trees in your zip code (using NYC Urban Forest OpenData).

CSci 127 (Hunter)

Lecture 13

5 December 2018 23 / 41

イロト 不良 トイヨト イヨト ヨー のくや

#Name: your name here
#Date: October 2017
#This program, uses functions,
says hello to the world!

def main():
 print("Hello, World!")

if __name__ == "__main__":
 main()

Plan: Alternate between working in pairs and sketching solutions (until time runs out):

```
//Another C++ program: Demonstrates loops
#include <iostream>
using namespace std:
int main ()
  int i.i.size:
  cout << "Enter size: ";
  cin >> size:
  for (i = 0; i < size; i ++)
  ł
    for (i = 0; i < size; i++)
     cout << "*":
    cout << endl:
  3
  cout << "\n\n":
  for (i = size; i > 0; i--)
    for (j = 0; j < i; j++)
     cout << "*";
    cout << endl;</pre>
 }
  return 0;
```

3

Sac

#Name: your name here
#Date: October 2017
#This program, uses functions,
says hello to the world!

```
def main():
    print("Hello, World!")
```

if __name__ == "__main__":
 main()

```
//Another C++ program: Demonstrates loops
#include <iostream>
using namespace std:
int main ()
  int i.i.size:
  cout << "Enter size: ";
  cin >> size:
  for (i = 0; i < size; i ++)
    for (j = 0; j < size; j ++)
    cout << "*":
    cout << endl:
  3
  cout << "\n\n":
  for (i = size; i > 0; i--)
    for (j = 0; j < i; j++)
    cout << "*";
    cout << endl;</pre>
  3
  return 0;
```

Plan: Alternate between working in pairs and sketching solutions (until time runs out):

• Definite Loops in Python & C++

5 December 2018 24 / 41

3

#Name: your name here
#Date: October 2017
#This program, uses functions,
says hello to the world!

```
def main():
    print("Hello, World!")
```

if __name__ == "__main__": main()

```
//Another C++ program: Demonstrates loops
#include <iostream>
using namespace std:
int main ()
  int i.i.size:
  cout << "Enter size: ";
  cin >> size:
  for (i = 0; i < size; i ++)
    for (j = 0; j < size; j ++)
     cout << "*":
    cout << endl:
  3
  cout << "\n\n":
  for (i = size; i > 0; i--)
    for (j = 0; j < i; j++)
    cout << "*";
    cout << endl;</pre>
  3
  return 0;
```

Plan: Alternate between working in pairs and sketching solutions (until time runs out):

- Definite Loops in Python & C++
- Conditionals in Python & C++

3

#Name: your name here
#Date: October 2017
#This program, uses functions,
says hello to the world!

```
def main():
    print("Hello, World!")
```

if __name__ == "__main__": main()

```
//Another C++ program: Demonstrates loops
#include <iostream>
using namespace std:
int main ()
  int i.i.size:
  cout << "Enter size: ";
  cin >> size:
  for (i = 0; i < size; i ++)
    for (j = 0; j < size; j ++)
     cout << "*":
    cout << endl:
  3
  cout << "\n\n";
  for (i = size; i > 0; i--)
    for (j = 0; j < i; j++)
     cout << "*";
    cout << endl;</pre>
  3
  return 0;
```

Plan: Alternate between working in pairs and sketching solutions (until time runs out):

- Definite Loops in Python & C++
- Conditionals in Python & C++
- Indefinite Loops in Python & C++

Complete as many as possible:

• Python: what is the output? for i in range(2017, 2000, -2): print("Year is", i)

• In Python, write a complete program that prints out 1 to 100.

• In C++, write a complete program that prints out 1 to 100.

CSci 127 (Hunter)

Lecture 13

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙

• Python: what is the output? for i in range(2017, 2000, -2): print("Year is", i)

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

• In Python, write a complete program that prints out 1 to 100.

• In C++, write a complete program that prints out 1 to 100.

- Write Python code that asks for the hour (24-hour time) and greets then with "Good Morning" if before 12, "Good Afternoon" for 12 but not yet 17, and "Good Evening" otherwise.
- 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)

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

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

• Write Python code that asks for the hour (24-hour time) and greets then with "Good Morning" if before 12, "Good Afternoon" for 12 but not yet 17, and "Good Evening" otherwise.

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~

 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).

In Pairs or Triples: Indefinite Loops in Python & C++ Complete as many as possible:

イロト 不良 トイヨト イヨト ヨー のくや

In Pairs or Triples: Indefinite Loops in Python & C++ Complete as many as possible:

```
• Python: what is the output?
  bal = 100
  while bal < 200:
       print("Balance", bal)
       bal = bal + 0.1*bal
• C++: what is the output?
  int n = 10;
  do {
       if (n \% 2 == 0)
          n = n / 2:
       else
          n = 3*n + 1;
       cout << "n is " << endl;</pre>
  } while (n > 1);
```

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

```
• Python: what is the output?
bal = 100
while bal < 200:
    print("Balance", bal)
    bal = bal + 0.1*bal
```

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

```
• C++: what is the output?
int n = 10;
do {
    if ( n % 2 == 0)
        n = n / 2;
    else
        n = 3*n + 1;
    cout << "n is " << endl;
} while (n > 1);
```

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

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

996

イロト 不得 トイヨト イヨト 二日

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

990

Practice Quiz & Final Questions



• Lightning rounds:

CSci 127 (Hunter)

Lecture 13

5 December 2018 40 / 41

3

990

Practice Quiz & Final Questions



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

イロト イロト イヨト イ

Practice Quiz & Final Questions



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

CSci 127 (Hunter)

Lecture 13

5 December 2018 40 / 41

イロト イロト イヨト イ
Practice Quiz & Final Questions



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

イロト イポト イヨト イヨト

Practice Quiz & Final Questions



- 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

イロト イポト イヨト イヨ

Practice Quiz & Final Questions



- 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)

Lecture 13

イロト イポト イヨト イヨ

Writing Boards



• Return writing boards as you leave...

CSci 127 (Hunter)

Lecture 13

5 December 2018 41 / 41

3

900

<ロト <回ト < 回ト < 回ト