## CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

< ロ ト < 回 ト < 三 ト < 三 ト</p>

CSci 127 (Hunter)

Lecture 13

December 6, 2022 1 / 44

3

990

• Please take a moment to fill out the Teacher Evaluations

990

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

= nar

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

イロト イポト イヨト イヨト

Э

Sac

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

3

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

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

#### • Final Exam December 19 at 9-11 AM

- Final Exam December 19 at 9-11 AM
- Seating assignment will be available on Blackboard/My Grades by this week
- Next Tuesday December 13, we will have a Mock Exam

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

- Final Exam December 19 at 9-11 AM
- Seating assignment will be available on Blackboard/My Grades by this week
- Next Tuesday December 13, we will have a Mock Exam
  - ► Room 118 Hunter North (Assembly Hall), ground floor of the North Building

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

- Final Exam December 19 at 9-11 AM
- Seating assignment will be available on Blackboard/My Grades by this week
- Next Tuesday December 13, we will have a Mock Exam
  - ► Room 118 Hunter North (Assembly Hall), ground floor of the North Building
  - ► Only 1.15 hours for the Mock, 2 hours for the real exam.

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙

- Final Exam December 19 at 9-11 AM
- Seating assignment will be available on Blackboard/My Grades by this week
- Next Tuesday December 13, we will have a Mock Exam
  - ► Room 118 Hunter North (Assembly Hall), ground floor of the North Building
  - ► Only 1.15 hours for the Mock, 2 hours for the real exam.
  - ► Just a practice run, this WILL NOT be the same as the real exam, and it will not be graded.

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

#### CSci 127: Introduction to Computer Science

CSci 127 (Hunter)

Lecture 13

= December 6, 2022 4 / 44

990

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

3

990

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

= nar

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

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

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

- What's the best way to study for the final exam? The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews. Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study. Deadline: Monday, December 5, 2022, 11:59 PM (EST)

イロト 不同下 イモト イモト

- What's the best way to study for the final exam? The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews. Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study. Deadline: Monday, December 5, 2022, 11:59 PM (EST)
- Why do you care about cheating?

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

 I'm worried about my grade. Should I do Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study. Deadline: Monday, December 5, 2022, 11:59 PM (EST)

 Why do you care about cheating? First: it gives unfair advantage & is immoral.

Sac

 What's the best way to study for the final exam? The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews. Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour

(half time) and work through , grade yourself, update note sheet, and repeat.

- I'm worried about my grade. Should I do Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study. Deadline: Monday, December 5, 2022, 11:59 PM (EST)
- Why do you care about cheating?
   First: it gives unfair advantage & is immoral.
   Second: it degrades the quality of our students.

 What's the best way to study for the final exam? The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews. Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour

(half time) and work through , grade yourself, update note sheet, and repeat.

- I'm worried about my grade. Should I do Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study. Deadline: Monday, December 5, 2022, 11:59 PM (EST)
- Why do you care about cheating?
   First: it gives unfair advantage & is immoral.
   Second: it degrades the quality of our students.
   Third: it's a standard question on faculty references.

Sac

 What's the best way to study for the final exam? The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews. Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour

(half time) and work through , grade yourself, update note sheet, and repeat.

- I'm worried about my grade. Should I do Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study. Deadline: Monday, December 5, 2022, 11:59 PM (EST)
- Why do you care about cheating? First: it gives unfair advantage & is immoral. Second: it degrades the quality of our students. Third: it's a standard question on faculty references. Industry & graduate schools hate it: don't want someone who falsifies work.

200

## 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: 2;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}</pre>
```

- Recap: I/O & Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python

Sac

## 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: 2;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}</pre>
```

- Recap: I/O & Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python

Sac

イロト 不得 トイヨト イヨト 二日

```
1 //C++ program demonstrating I/O & arithmetic
2 #include <iostream>
 using namespace std;
3
4
  int main ()
5
  ł
6
     float kg, lbs;
7
     cout << "Enter kg: ";</pre>
8
     cin >> kg;
9
     lbs = kg * 2.2;
10
     cout << endl << "Lbs: " << lbs << "\n\n":
11
     return 0;
12
13
                                                     = 090
    CSci 127 (Hunter)
                           Lecture 13
                                              December 6, 2022
                                                        8 / 44
```

• 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;
```

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:

```
//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;
```

Sac

- 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;
```

3

Sac

- 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;
```

3

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

1	<pre>//Another C++ program, demonstrating I/O &amp; arithmetic finclude siostream&gt; using namespace std;</pre>
1	nt main O { float kg, lbs; cout << "Enter kg; "; cin >> kg; lbs = kg * 2.2; cout << endl << "Lbs; " << lbs << "\n\n"; return 0;
13	

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

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

To print:

E 990

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

#### • Efficient for systems programming.

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

int, float, char, ...

• To print: cout << "Hello!!";

3

3

200

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

3

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

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

int, float, char, ...

- To print: cout << "Hello!!";
- To get input: cin >> num;

3

Sac

<pre>//Another C++ program, demonstrating 1/0 &amp; arithme #include <iostream> using namespace std;</iostream></pre>
$\label{eq:constraint} \begin{array}{l} \text{int main } O \\ \{ \text{fott } k_0, \text{ bs}; \\ \text{cont} < < \text{"Enter } k_0; \; \text{"}; \\ \text{cin } > k_0; \\ \text{lbs } = k_0 \neq 2.2; \\ \text{cont} < < \text{end} < < \text{"Lbs}; \; \text{"} < \text{ lbs } < < \text{"\n'\n"}; \\ \text{return } 0; \\ \text{return } 0; \\ \end{array}$

- 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 1/0 &amp; arithm #include <iostream> using namespace std;</iostream></pre>	ne
<pre>int main () {     float kg, lbs;     cout &lt;&lt; "Enter kg: ";</pre>	
cin >> kg;	
$cout < cout < "lbs: " < lbs < "\n\n":$	
noture At	

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, #include <iostream> using namespace std;</iostream></pre>	demonstrating	1/0 &	arithme
<pre>int main () {  float kg, lbs;  cout &lt;&lt; "Enter kg: "  cin &gt;&gt; kg;  lbs = kg * 2.2;  cout &lt;&lt; endl &lt;&lt; "Lbs  return 0;</pre>	; = << lbs << '	"\n\n"	;

- 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 &amp; arithme #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";
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>

<pre>//Another C++ program, demonstrating I/O &amp; 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++) {...}</pre>
- Blocks of code uses '{' and '}'.

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~

<pre>//Another C++ program, demonstrating I/O &amp; 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
#include <lostream>
using namespace std;
```

```
int main ()
```

```
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg: 2;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}</pre>
```

- Recap: I/O & Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python

= nar

# Challenge:

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

December 6, 2022 11 / 44

= nar

#### C++ Demo, https://www.onlinegdb.com/Hk\_q0V8xf

```
//C++ program demonstrates conditionals
#include <iostream>
using namespace std;
```

```
int main ()
ſ
    int yearBorn;
    cout << "Enter year born: ";</pre>
    cin >> yearBorn;
    if (yearBorn < 1946)
    Ł
        cout << "Greatest Generation";</pre>
    3
    else if (yearBorn <= 1964)
        cout << "Baby Boomer":</pre>
    else if (vearBorn <= 1984)
        cout << "Generation X";</pre>
    else if (vearBorn <= 2004)
    Ł
        cout << "Millennial";</pre>
    }
    else
        cout << "TBD";</pre>
    }
    return 0;
```

1

2

3

4 5

3

Sac

# Conditionals

#### General format:



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

CSci 127 (Hunter)

Lecture 13

December 6, 2022 13 / 44

3

Sac

## Logical Operators in C++

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

<□> <同> <同> < 回> < 回> < 回> < 回> < 回> < 0 < 0

## Logical Operators in C++ $\,$

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

and (&&)

in1		in2	returns:
False	&&	False	False
False	&&	True	False
True	&&	False	False
True	&&	True	True

#### Logical Operators in C++

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

#### and (&&)

in1		in2	returns:
False	&&	False	False
False	&&	True	False
True	&&	False	False
True	&&	True	True

or (||)

in1	in2	returns:
False	False	False
False	True	True
True	False	True
True	True	True

#### Logical Operators in C++

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

#### and (&&)

in1		in2	returns:
False	&&	False	False
False	&&	True	False
True	&&	False	False
True	&&	True	True

or (||)

in1	in2	returns:
False	False	False
False	True	True
True	False	True
True	True	True

not (!)

	in1	returns:
!	False	True
!	True	False

CSci 127 (Hunter)

December 6, 2022 14 / 44

# Lecture Slip

• Write a C++ program that will ask for the time in 24 hour format and, knowing it is morning before 12pm and evening after 6pm (18), it will print out Morning, Afternoon or Evening.

Sac

#### Today's Topics

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

```
int main ()
```

```
  float kg, lbs;
  cout << "Enter kg: ";
  cin >> kg: 2;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}</pre>
```

- Recap: I/O & Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python

200

イロト 不得 トイヨト イヨト 二日

Challenge: predict what code will do: https://onlinegdb.com/kUuuLlejO

```
//While Growth example
1
   #include <iostream>
2
   using namespace std;
3
4
   int main ()
5
   ł
6
     int population = 100;
7
     int year = 0;
8
     cout << "Year\tPopulation\n";</pre>
9
     while (population < 1000)
10
     ſ
11
          cout << year << "\t" << population << "\n";</pre>
12
          population = population * 2;
13
          year++;
14
     }
15
     return 0;
16
                                                               =
                                              イロト イポト イヨト イヨト
17
     CSci 127 (Hunter)
                                Lecture 13
                                                      December 6, 2022
```

nak

17 / 44

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

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

link: https://onlinegdb.com/kUuuLlej0

CSci 127 (Hunter)

Sac

#### Indefinite Loops: while

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

General format:

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

command1; command2; command3;

3

Sac

# Challenge: predict what the code do

```
#include <iostream>
1
   using namespace std;
2
3
   int main ()
4
5
      int num;
6
      cout << "Enter an even number: ";</pre>
7
      cin >> num;
8
      while (num % 2 != 0)
9
      ł
10
          cout << "\nThat's odd!\n";</pre>
11
          cout << "Enter an even number: ";</pre>
12
          cin >> num;
13
     }
14
      cout << "You entered: " << num << ".\n";</pre>
15
      return 0;
16
17
      CSci 127 (Hunter)
                                    Lecture 13
                                                            December 6, 2022
```

Sale

20 / 44

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

```
//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 ∅;
}
```

https://www.onlinegdb.com/rJttLSLgG

= nar

#### 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 0;
```

General format:

```
while ( logical expression )
```

command1; command2; command3;

. . .

= nar

Challenge: predict what the code will do

```
//Demonstrates do-while loops
1
   #include <iostream>
2
   using namespace std;
3
4
   int main ()
5
   ſ
6
     int num;
7
     do
8
     ſ
9
          cout << "Enter an even number: ";</pre>
10
         cin >> num:
11
     } while (num % 2 != 0);
12
13
     cout << "You entered: " << num << ".\n";</pre>
14
     return 0;
15
16
```

## C++ Demo:

```
//Demonstrates do-while loops
   #include <iostream>
   using namespace std;
   int main ()
   Ł
     int num;
     do
                                             (Demo with onlinegdb)
     {
         cout << "Enter an even number: ";</pre>
        cin >> num;
     } while (num % 2 != 0);
     cout << "You entered: "
         << num << ".\n";
     return 0:
   3
link: https://www.onlinegdb.com/Bkn8DB8eG
```

= 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:
}
```

Lecture 13

= 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: 2;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n";
  return 0;
}</pre>
```

- Recap: I/O & Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python

Sac

• I/O:

• I/O: cin >> ...;

Lecture 13

December 6, 2022 27 / 44

• 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|] << endt;

return 0;

} return 0;
```

• I/O: cin >> ...; & cout << ...;

• Definite loops:



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

```
 \begin{array}{l} (\text{in that } O \\ \{ \text{ ort } i, j \} \\ \text{for } (1 - 0; \ i < 4; \ i + ) \\ f \\ \text{for } (1 - 0; \ i < 4; \ i + ) \\ \text{for } (j = 10; \ j > 0; \ j - .) \\ \text{for } (j = 10; \ j > 0; \ j - .) \\ \text{four } (s - s \ s < " \ "; \ s \\ \text{cont} < \text{``Shart off}(1 \ \ \text{end}); \\ \text{return } 0; \\ \end{array}
```

Jac.

イロト 不得 トイヨト イヨト 二日

```
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 that } O \\ & (\text{in } i,j); \\ (\text{in } i,j); \\ (\text{or } (j=0;\ i<4;\ i**) \\ & (\text{or } (j=0;\ i<4;\ i**) \\ & (\text{or } (j=10;\ j>0;\ j-\cdot) \\ & (\text{out } \ll i \neq < ""; \\ & (\text{out } \ll i \text{Blast off}!) \ll \text{end}; \\ & (\text{return } 0; \\ \end{array}
```

= nar

```
#include clostroom=
using memory std;
for (j = 0; i < 4; i+=)
for (j = 10; i < 4; i+=)
for (j = 10; j > 0; j --)
{ cout << 'line sort/d turned upside down...\vi';
for (j = 10; j > 0; j --)
{ cout << 'line sort/d'urned upside down...\vi';
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
```

3

Sac

```
\label{eq:constraints} C+i program; Demonstrates loops \\ \mbox{wing manapace std;} \\ \mbox{wing manapace std;} \\ \mbox{wing manapace std;} \\ \mbox{in tenin } 0 \\ \left\{ \begin{array}{l} \mbox{int tenin } 0 \\ \mbox{in tenin } 0 \\ \mbo
```

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

Lecture 13

December 6, 2022 27 / 44

3

Sac

```
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)
                                                                ...
```

CSci 127 (Hunter)

Lecture 13

Definite loops:

I/O: cin >> ...; & cout << ...;
</pre>
• 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;
}</pre>
```

CSci 127 (Hunter)

3

28 / 44

December 6, 2022

< □ > < □ > < □ > < □ >

• *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()
{
```

```
• 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 SQC

```
• 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>
```

CSci 127 (Hunter)

200

```
• 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)

Sac

• 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>
```

- 4 A b

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):
```

CSci 127 (Hunter)

• 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)

December 6, 2022 30 / 44

Leap Year

A year is a leap year if it is divisible by 4, but century years are not leap years unless they are divisible by 400.

rectangle represents all the years





 inner red circle means years divided by 400. It is completely inside yellow circle.

• red shape: leap year, yellow shape: non-leap year.

- (1) Red ring represents (year % 4 == 0) and not (year % 100 == 0).
- ② Inner red circle represents (year % 400 == 0).
- 3 Red shape represents ((year % 4 == 0) and not (year % 100 == 0)) or (year % 400 == 0), same as (year % 4 == 0) and ( not (year % 100 == 0) or (year % 400 == 0) ) by De Morgan's Law.

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!!")
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 \$33.00, and the cost of single ride, with bonus, is \$2.75).

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!!")
print("Year")
```

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

```
if TRUE and \
    (not (year % 100 == 0) or (year % 400 == 0)):
    print("Leap!!")
print("Year")
```

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!!")
print("Year")
```

▲□▶ ▲□▶ ▲三▶ ▲三▶ 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")
```

```
year = 2016
if TRUE and \
    (not FALSE or (year % 400 == 0)):
        print("Leap!!")
print("Year")
```

CSci 127 (Hunter)

December 6, 2022 34 / 44

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

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

```
year = 2016
if TRUE and \
    (TRUE or (year % 400 == 0)):
    print("Leap!!")
print("Year")
```

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!!")
print("Year")
```

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

```
year = 2016
if TRUE and \
    (TRUE or FALSE):
    print("Leap!!")
print("Year")
```

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!!")
print("Year")
```

```
year = 2016
if TRUE and \
    (TRUE or FALSE):
    print("Leap!!")
print("Year")
```

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!!")
print("Year")
```

```
year = 2016
if TRUE and \
    (TRUE):
    print("Leap!!")
print("Year")
```

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!!")
print("Year")
```

```
year = 2016
if TRUE:
    print("Leap!!")
print("Year")
```

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

```
year = 2016
if TRUE:
    print("Leap!!")
print("Year")
```

```
Prints: Leap!
Year
CSci 127 (Hunter) Lecture 13 December 6, 2022 39 / 44
```

 Your program should then print if it is cheaper to buy single ride metro cards (\$2.75 per ride) or 7-day unlimited card (\$33.00).
 #include <iostream> using namespace std;

200

イロト 不得 トイヨト イヨト 二日

 Your program should then print if it is cheaper to buy single ride metro cards (\$2.75 per ride) or 7-day unlimited card (\$33.00).
 #include <iostream> using namespace std; int main()

200

イロト 不得 トイヨト イヨト 二日

 Your program should then print if it is cheaper to buy single ride metro cards (\$2.75 per ride) or 7-day unlimited card (\$33.00).
 #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
 (\$2.75 per ride) or 7-day unlimited card (\$33.00).
 #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
 (\$2.75 per ride) or 7-day unlimited card (\$33.00).
 #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
 (\$2.75 per ride) or 7-day unlimited card (\$33.00).
#include <iostream>
using namespace std;
int main()
 {
 int rides;
 cout << "Enter number of rides:";
 cin >> rides;
 if (2.75 \* rides < 33.00)</pre>

200

• Your program should then print if it is cheaper to buy single ride metro cards (\$2.75 per ride) or 7-day unlimited card (\$33.00). #include <iostream> using namespace std; int main() int rides; cout << "Enter number of rides:";</pre> cin >> rides; if (2.75 \* rides < 33.00)cout << "Cheaper to buy single ride metro cards.\n";

• Your program should then print if it is cheaper to buy single ride metro cards (\$2.75 per ride) or 7-day unlimited card (\$33.00). #include <iostream> using namespace std; int main() int rides; cout << "Enter number of rides:";</pre> cin >> rides; if (2.75 \* rides < 33.00)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 (\$2.75 per ride) or 7-day unlimited card (\$33.00). #include <iostream> using namespace std; int main() int rides; cout << "Enter number of rides:";</pre> cin >> rides; if (2.75 \* rides < 33.00)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
  ($2.75 per ride) or 7-day unlimited card ($33.00).
  #include <iostream>
  using namespace std;
  int main()
    int rides;
    cout << "Enter number of rides:";</pre>
    cin >> rides;
    if (2.75 * rides < 33.00)
      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
                                                       December 6, 2022
                                                                     40 / 44
```

• 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

December 6, 2022 41 / 44

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙
• Write Python code that repeatedly prompts for a non-empty string.

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

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

s = ""

• 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.

```
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.

```
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;

• 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)

• 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)

• 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;
```

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;

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; ▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~ CSci 127 (Hunter) Lecture 13 December 6, 2022 42 / 44



Before next lecture, don't forget to:

• Work on this week's Online Lab

Э

590

イロト イロト イヨト



Before next lecture, don't forget to:

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

CSci 127 (Hunter)

Lecture 13

December 6, 2022 43 / 44



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001G Hunter North
- Submit this week's 5 programming assignments (programs 55-60)

CSci 127 (Hunter)



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001G Hunter North
- Submit this week's 5 programming assignments (programs 55-60)
- If you need help, schedule an appointment for Tutoring in lab 1001G 11:30am-5pm

CSci 127 (Hunter)



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001G Hunter North
- Submit this week's 5 programming assignments (programs 55-60)
- If you need help, schedule an appointment for Tutoring in lab 1001G 11:30am-5pm
- Take the Lecture Preview on Blackboard on Monday (or no later than 10:15am on Tuesday)

CSci 127 (Hunter)

# Lecture Slips & Writing Boards



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

CSci 127 (Hunter)

Lecture 13

December 6, 2022 44 / 44

-