CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

< □ > < □ > < □ > < □ > < □ >

CSci 127 (Hunter)

Lecture 13

3 8 May 2018 1 / 44

990

From lecture slips & recitation sections.

• Who/why all the visitors?

From lecture slips & recitation sections.

Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.

イロト 不良 トイヨト イヨト ヨー のくや

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit?

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit? *Programs are worth 30% of the final grade.*

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit? Programs are worth 30% of the final grade. Lecture slips & in-class quizzes can only help you (up to 35% of your grade).

Sac

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit? Programs are worth 30% of the final grade. Lecture slips & in-class quizzes can only help you (up to 35% of your grade).
 - No programs & 100% on final (adds 70%) \Rightarrow 70%.

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit? Programs are worth 30% of the final grade. Lecture slips & in-class quizzes can only help you (up to 35% of your grade).
 - No programs & 100% on final (adds 70%) \Rightarrow 70%.
 - $\frac{1}{3}$ programs (10%) & 100% on final (70%) \Rightarrow 80%.

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit? Programs are worth 30% of the final grade. Lecture slips & in-class quizzes can only help you (up to 35% of your grade).
 - No programs & 100% on final (adds 70%) \Rightarrow 70%.
 - $\frac{1}{3}$ programs (10%) & 100% on final (70%) \Rightarrow 80%.
 - All programs (30%) & 60% on final (42%) ⇒ 72%.

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit? Programs are worth 30% of the final grade. Lecture slips & in-class quizzes can only help you (up to 35% of your grade).
 - No programs & 100% on final (adds 70%) \Rightarrow 70%.
 - $\frac{1}{3}$ programs (10%) & 100% on final (70%) \Rightarrow 80%.
 - All programs (30%) & 60% on final (42%) \Rightarrow 72%.

We also give Credit/NoCredit- check with your advisor to make sure it's accepted for your program of study.

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit? Programs are worth 30% of the final grade. Lecture slips & in-class quizzes can only help you (up to 35% of your grade).
 - No programs & 100% on final (adds 70%) \Rightarrow 70%.
 - $\frac{1}{3}$ programs (10%) & 100% on final (70%) \Rightarrow 80%.
 - All programs (30%) & 60% on final (42%) \Rightarrow 72%.

We also give Credit/NoCredit- check with your advisor to make sure it's accepted for your program of study.

I want to learn more- what should I take next?

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit? Programs are worth 30% of the final grade. Lecture slips & in-class quizzes can only help you (up to 35% of your grade).
 - No programs & 100% on final (adds 70%) \Rightarrow 70%.
 - $\frac{1}{3}$ programs (10%) & 100% on final (70%) \Rightarrow 80%.
 - All programs (30%) & 60% on final (42%) \Rightarrow 72%.

We also give Credit/NoCredit- check with your advisor to make sure it's accepted for your program of study.

- I want to learn more- what should I take next?
 - ► Majors: CSci 135/136 (C++, MTh 8:10-9:25 & section) & CSci 150 (Discrete Structures, MTh 1:10-2:25pm & section)

CSci 127 (Hunter)

8 May 2018 2 / 44

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit? Programs are worth 30% of the final grade. Lecture slips & in-class quizzes can only help you (up to 35% of your grade).
 - No programs & 100% on final (adds 70%) \Rightarrow 70%.
 - $\frac{1}{3}$ programs (10%) & 100% on final (70%) \Rightarrow 80%.
 - All programs (30%) & 60% on final (42%) \Rightarrow 72%.

We also give Credit/NoCredit- check with your advisor to make sure it's accepted for your program of study.

• I want to learn more- what should I take next?

- ► Majors: CSci 135/136 (C++, MTh 8:10-9:25 & section) & CSci 150 (Discrete Structures, MTh 1:10-2:25pm & section)
- Minors: CSci 133 (More Python: multiple times) & CSci 232 (Databases, multiple times)

CSci 127 (Hunter)

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit? Programs are worth 30% of the final grade. Lecture slips & in-class quizzes can only help you (up to 35% of your grade).
 - No programs & 100% on final (adds 70%) \Rightarrow 70%.
 - $\frac{1}{3}$ programs (10%) & 100% on final (70%) \Rightarrow 80%.
 - All programs (30%) & 60% on final (42%) \Rightarrow 72%.

We also give Credit/NoCredit- check with your advisor to make sure it's accepted for your program of study.

I want to learn more- what should I take next?

- ► Majors: CSci 135/136 (C++, MTh 8:10-9:25 & section) & CSci 150 (Discrete Structures, MTh 1:10-2:25pm & section)
- Minors: CSci 133 (More Python: multiple times) & CSci 232 (Databases, multiple times)
- What's a mock exam? I see it on the webpage...

CSci 127 (Hunter)

From lecture slips & recitation sections.

- Who/why all the visitors?
 We're part of a pilot program for prospective students. Last visit is today.
- I'm worried about my grade. Should I do Credit/NoCredit? Programs are worth 30% of the final grade. Lecture slips & in-class quizzes can only help you (up to 35% of your grade).
 - No programs & 100% on final (adds 70%) \Rightarrow 70%.
 - $\frac{1}{3}$ programs (10%) & 100% on final (70%) \Rightarrow 80%.
 - All programs (30%) & 60% on final (42%) \Rightarrow 72%.

We also give Credit/NoCredit- check with your advisor to make sure it's accepted for your program of study.

• I want to learn more- what should I take next?

- ► Majors: CSci 135/136 (C++, MTh 8:10-9:25 & section) & CSci 150 (Discrete Structures, MTh 1:10-2:25pm & section)
- Minors: CSci 133 (More Python: multiple times) & CSci 232 (Databases, multiple times)

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

▲□▶ ▲□▶ ▲ヨ▶ ▲ヨ▶ ヨ - のの⊙

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

E ∽QQ

• Efficient for systems programming.

```
//Another C++ program, demonstrating I/O & arithmetic
minuled sitestream-
using namespace std;
int main O
{ float kg, lbs;
cout << "Enter kg: ";
lbs = kg * 2.2;
cout << endl << "lbs: " << lbs << "\n\n";
return 0;
}
```

8 May 2018 5 / 44

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

```
//Another C++ program, demonstrating I/O & arithmetic
#include ciostream.
using namespace std;
int main O
{
float kg, lbs;
cout << "Enter kg: ";
lbs - kg * 2.2;
cout << endl << "lbs: " << lbs << "\n\n";
}
```

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.
- 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 cisteream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "Enter kg: ";
  cin > kg; 2.2;
  coat << endl << "Lbs: " << lbs << "\n\n";
  return 0;
```

200

- 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 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

```
//Another C++ program, demonstrating I/O & arithmetic
finclude <lostreams
using namespace std;
int main O
{
float kg, lbs;
coat << "Enter kg: ";
tis kg, 2.2;
tis kg, 2.2;
coat << endl << "Lbs: " << lbs << "\n\n";
}
eturn 0;
}
```

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

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

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
wining ammespace std;
int main ()
{ float kg, lbs;
coat >< driver kg; ";
coat >< kg; ?;
lbs = kg ? 2.2;
coat << mg; * lbs << "\n\n";</pre>
```

return 0;
}

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

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

To print:

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cistoream.
using namespace std;
int main ()
{ float kg, lbs;
coat << "inter kg: ";
tos kg ? 2.2;
coat << "inter kg: " << lbs << "\\n\n";</pre>
```

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!!";

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cin > 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;
- Many types available:

int, float, char, ...

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

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cin > 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;
- Many types available: int, float, char, ...
- To print: cout << "Hello!!";
- To get input: cin >> num;

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cin > 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;
- Many types available:

int, float, char, ...

- To print: cout << "Hello!!";
- To get input: cin >> num;
- To use those I/O functions:

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
#include :clostream.
using manespace std;
int main O
{
foot kg, lbs;
cout << "Enter kg: ";
cot << "Enter kg: ";
cot << endl.cl";
cot <</pre>
```

3

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

using namespace std;

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
#include :clostream.
using manespace std;
int main O
{
foot kg, lbs;
cout << "Enter kg: ";
cot << "Enter kg: ";
cot << endl.cl";
cot <</pre>
```

3

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

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
minuled ciostreams
using namespace std;
int main O
{
foot kg, lbs;
cout << "Enter kg: ";
cio >> kg;
cout << email.c?;
cout << cout </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>

イロト イポト イヨト イヨト 二日

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
%include ciostreams
using namespace std;
int main O
{
foot kg, lbs;
cout << "Enter kg: ";
cin >> kg' 2.2;
lbs - kg * 2.2;
return @ eff( <* lbs * < "\n\n";
return % of *;
```

- 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>

イロト イポト イヨト イヨト 二日

• Blocks of code uses '{' and '}'.

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
%include ciostreams
using namespace std;
int main O
{
foot kg, lbs;
cout << "Enter kg: ";
cin >> kg' 2.2;
lbs - kg * 2.2;
return @ eff( <* lbs * < "\n\n";
return % of *;
```

- 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 ';'.

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

Side Note: gdb

• Part of Richard Stallman's "GNU is Not Unix" (GNU) project.



gdb.org

CSci 127 (Hunter)

Lecture 13

3 8 May 2018 6 / 44

200

Side Note: gdb



gdb.org

- Part of Richard Stallman's "GNU is Not Unix" (GNU) project.
- Written in 1986, gdb is the GNU debugger and based on dbx from the Berkeley Distribution of Unix.

Side Note: gdb



gdb.org

- Part of Richard Stallman's "GNU is Not Unix" (GNU) project.
- Written in 1986, gdb is the GNU debugger and based on dbx from the Berkeley Distribution of Unix.
- Lightweight, widely-available program that allows you to "step through" your code line-by-line.

A (1) < </p>

Side Note: gdb



gdb.org

- Part of Richard Stallman's "GNU is Not Unix" (GNU) project.
- Written in 1986, gdb is the GNU debugger and based on dbx from the Berkeley Distribution of Unix.
- Lightweight, widely-available program that allows you to "step through" your code line-by-line.
- Available on the lab machines (via command-line and the IDE spyder) and on-line (onlinegdb.com).

In Pairs or Triples:

Predict what the following pieces of code will do:

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

```
using namespace std;
int main ()
£
    string conditions = "blowing snow";
    int winds = 100;
    float visibility = 0.2;
    if ( ( (winds > 35) && (visibility < 0.25) )
         ( (conditions == "blowing snow") ||
           (conditions == "heavy snow") ) )
        cout << "Blizzard!\n":</pre>
    string origin = "South Pacific";
    if (winds > 74)
        cout << "Major storm, called a ";</pre>
    if ((origin == "Indian Ocean")
        ||(origin == "South Pacific"))
        cout << "cyclone.\n";</pre>
    else if (origin == "North Pacific")
        cout << "typhoon.\n";</pre>
    else
        cout << "hurricane.\n";</pre>
```

イロト イポト イヨト イヨト

```
CSci 127 (Hunter)
```

Lecture 13

3 8 May 2018 7 / 44

Sac

C++ Demo

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

CSci 127 (Hunter)

3 8 May 2018 8 / 44

Sac

Conditionals

General format:



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

CSci 127 (Hunter)

3 8 May 2018 9 / 44

Sar

Logical Operators in C++

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

<ロト < 部 ト < 注 ト < 注 ト 三 三 の < ()</p>

Logical Operators in C++ $\,$

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

and (&&)

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

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)

8 May 2018 10 / 44

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)

8 May 2018 10 / 44

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

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
                                                 8 May 2018
```

11 / 44

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

```
//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 < 'N000)
    {
        cout << year << "\t" << population << '\n";
    }
    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 0;
```

CSci 127 (Hunter)

Sac

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

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

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

Indefinite Loops: while

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

General format:

```
while ( logical expression )
```

command1; command2; command3;

. . .

= nar

In Pairs or Triples:

```
Predict what the following pieces of code will do:
 //Demonstrates do-while loops
 #include <iostream>
 using namespace std;
 int main ()
  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)

8 May 2018 17 / 44

E Sac

C++ 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 ∅;
}
```

Sac

<ロト <回ト < 回ト < 回ト

Indefinite Loops: do-while

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

= nar

In Pairs or Triples:

Predict what the following pieces of code will do:

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

= nar

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

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

∃ <\0<</p>

● I/O:

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

with monospore txd;

in team ()

{ mit j;

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

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

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

{ cout << j <= ";

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

} return 0;

}
```

• I/O: cin >> ...;

• I/O: cin >> ...; & cout << ...;

▲□▶ ▲□▶ ▲ヨ▶ ▲ヨ▶ ヨ - のの⊙

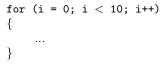
• I/O: cin >> ...; & cout << ...;

• Definite loops:

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

• I/O: cin >> ...; & cout << ...;

• Definite loops:



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

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

200

イロト 不得 トイヨト イヨト 二日

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

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

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

Sac

```
#include clostroom-
using memory std;
int min 0,
{
    for (j = 0; i < 4; i+-)
    for (j = 10; j > 0; j--)
    for (j = 10; j > 0; j--)
    for (j = 10; j < "";
    for cost <<" Sits off|)" << endl;
    return 0;
}
```

//Another C++ program: Demonstrates loops

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

3

Sac

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

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

3 8 May 2018 22 / 44

Sac

```
//decider (... program; Demonstrates loops finctule: -(dofreem) using memospace std; (int i,j; for (i = 0; t < 4; i+=) { ( cut < c ~ The world turned upside dom..., \n"; } ( cut < c ~ j < c ~ ; ... ) { cut < c > c < t < c ~ ; } cut < c > c < t < c ~ ; } ; ... ) { cut < c > c < t < c ~ ; } ; ... ) { cut < c > c < t < c ~ ; } ; ... ) { cut < c > c < t < c ~ ; } ; ... ) { cut < c > c < t < c ~ ; } ; ... ) { cut < c > c < t < c ~ ; } ; ... ) { cut < c > c < t < c ~ ; } ; ... ) { cut < c > c < t < c ~ ; } ; ... ) { cut < c > c < t < c ~ ; } ; ... ) { cut < c > c < t < c ~ ; } ; ... ) { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; } ; ... ] { cut < c > c < t < c ~ ; ] { cut < c > c < t < c ~ ; ] { cut < c > c < t < c ~ ; ] { cut < c > c < t < c ~ ; ] { cut < c > c < c ~ ; ] { cut < c > c ~ ; ] { cut < c ~ ; ] { cut <
```

CSci 127 (Hunter)

```
• I/O: cin >> ...; & cout << ...;</pre>
Definite loops:
  for (i = 0; i < 10; i++)
        ...
Conditionals:
  if (logical expression)
  else
Indefinite loops:
  while (logical expression)
        ...
                      イロト イポト イヨト イヨト
                                             3
                                                Sac
     Lecture 13
                                     8 May 2018
                                                22 / 44
```

Design Questions & Lecture Slips



• Requests: More on design questions & the mock final.

Design Questions & Lecture Slips



• Requests: More on design questions & the mock final.

Image: A math a math

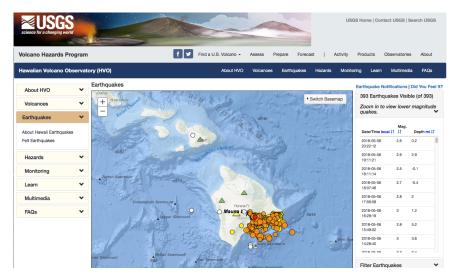
• Extra lecture slip today for design questions.

Design Questions & Lecture Slips



- Requests: More on design questions & the mock final.
- Extra lecture slip today for design questions.
- Only need to submit the slip with your name & ID (survey of programs).

Image: A math a math

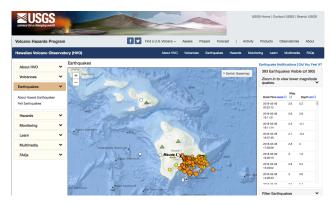


(USGS Volcano Hazards Program, May 2018)

CSci 127 (Hunter)

Lecture 13

<□▶ < □▶ < □▶ < 三▶ < 三▶ < 三 少 Q (~ 8 May 2018 24 / 44



(USGS Volcano Hazards Program, May 2018)

- Design an algorithm that maps the earthquakes in a USGS CSV file (using turtles or folium).
- Extra Challenge: Highlight the earthquake with maximum magnitude.

CSci 127 (Hunter)

1	A	В	C	D	E	F	G	н		J	K	L	M
1	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net	id	updated
2	2013-04-11T	20.7915	122.226	8.29	4.6	mb	46	115	2.28	1.21	us	usb000g50m	2013-04-11T
3	2013-04-11T	-17.3579	175.0663	9.88	5.3	mb		50	7.802	1.44		usb000g4z6	2013-04-11T
4	2013-04-11T	-17.4508	-178.7735	535.92	4.9	mb	21	128	8.56	0.71	us	usb000g4xf	2013-04-11T
5	2013-04-11T	-16.9546	-179.1921	528.46	4.5	mb	45	75	9.06	0.84	us	usb000g4ug	2013-04-11T
6	2013-04-11T	-10.6708	166.0755	167.21	4.7	mb	42	102	6.16	0.8	us	usb000g4su	2013-04-11T
7	2013-04-11T	2.8643	125.4971	64.96	4.6	mb		72	2.794	0.78	us	usb000g4rg	2013-04-11T
8	2013-04-11T	20.9199	122.1061	12.94	4.5	mb	32	130	2.12	0.69	us	usb000g4qu	2013-04-11T
9	2013-04-111	-2.7939	148.1628	9.39	4.6	mb	23	147	1.09	1.12	us	usb000g4g9	2013-04-11T
10	2013-04-11T	19.2629	95.6948	10.07	5.2	Mwp	75	43	2.14	0.98	us	usb000g4nw	2013-04-11T
11	2013-04-111	41.6165	141.9924	55.27	4.6	mb	43	123	0.96	1.09	us	usb000g4ni	2013-04-11T
12	2013-04-11T	28.5074	51.6758	10.07	4.8	mb		64	10.83	1.19	us	usb000g4mt	2013-04-11T
13	2013-04-10T	18.854	97.5096	8.27	4.7	mb	29	75	0.63	0.6	us	usb000g4i5	2013-04-11T
14	2013-04-10T	20.8187	122.1203	4.2	5.8	Merer	115	31	2.21	1.28	us	usb000g4ca	2013-04-10T
15	2013-04-10T	2.6017	127.2174	66.02	5	mb	60		1.82	1.03	us	usb000g4br	2013-04-10T
16	2013-04-10T	15.5366	-87.228	10	5.5	mb		37	1.471	0.85	us	usb000g4a2	2013-04-11T
17	2013-04-10T	-10.7302	-75.2622	99.62		mb		72	1.99	0.77	us	usb000g43v	2013-04-10T
18	2013-04-10T	-17.7569	167.7868	10	4.6	mb	20	160	3.71171	0.87	us	us2013nvap	2013-04-10T
19	2013-04-10T	28.5135	51.5523	9.93	4.6			94	10.877	0.97	us	usb000g3y3	2013-04-10T
20	2013-04-10T	28.438	51.738	9.87	5.2	mb	76	58	10.87	1.03	us	usb000g3ts	2013-04-10T
21	2013-04-10T	28.309	51.7514	10.06	4.8	mb		75	10.974	0.95	us	usb000g3t2	2013-04-10T
22	2013-04-10T	37.4728	142.0723	27.79	4.6	mb	40	132	3.23	1.17	us	usb000g3ge	2013-04-10T
23	2013-04-10T	28.45	51.6075	10.02	5.6	mb	76	25	10.91	1.14	us	usb000g3p7	2013-04-10T
24	2013-04-10T	-2.9729	139.0662	55.19	4.8	mb	33	61	6.81	1.6	us	usb000g3ns	2013-04-10T
25	2013-04-10T	28.4814	51.604	10	4.9	mb		139	10.883	0.83	us	usb000g3nn	2013-04-10T
26	2013-04-10T	-2.0824	-79.5666	103.34	4.5	mb	35	113	2.56	0.5	us	usb000g3ng	2013-04-11T
27	2013-04-09T	-22.7541	69.1376	10.2	4.6	mb	17	110	11.11	0.55	us	usb000g3ls	2013-04-09T
28	2013-04-09T	28.2759	51.6754	9.88	4.8	mb		86	11.034	0.68	us	usb000g3ft	2013-04-09T
29	2013-04-09T	5.6129	93.3101	31.21		mb		139	3.641	0.61	US	usb000g3fe	2013-04-09T
30	2013-04-09T	28,4201	51.6408	19.93	4.6	mb	94	94	10.92	0.72	us	usb000g3dn	2013-04-09T

(USGS Volcano Hazards Program, May 2018)

8 May 2018 26 / 44

<ロト < 部 ト < 注 ト < 注 ト 三 三 の < ()</p>

1	A	В	C	D	E	F	G	Н	1	J	K	L	M
1	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net	id	updated
2	2013-04-11T	20.7915	122.226	8.29	4.6	mb	46	115	2.28	1.21	us	usb000g50m	2013-04-111
3	2013-04-11T	-17.3579	175.0663	9.88	5.3	mb		50	7.802	1.44	us	usb000g4z6	2013-04-111
4	2013-04-11T	-17.4508	-178.7735	535.92	4.9	mb	21	128	8.56	0.71	us	usb000g4xf	2013-04-111
5	2013-04-11T	-16.9546	-179.1921	528.46	4.5	mb	45	75	9.06	0.84	us	usb000g4ug	2013-04-111
6	2013-04-111		166.0755	167.21		mb	42	102	6.16	0.8		usb000g4su	2013-04-111
7	2013-04-11T	2.8643	125.4971	64.96	4.6	mb		72	2.794	0.78	us	usb000g4rg	2013-04-111
8	2013-04-111		122.1061	12.94		mb	32	130	2.12			usb000g4qu	2013-04-111
9	2013-04-111	-2.7939	148.1628	9.39	4.6	mb	23	147	1.09	1.12	us	usb000g4q9	2013-04-111
10	2013-04-111		95.6948	10.07		Mwp	75	43	2.14			usb000g4nw	2013-04-111
11	2013-04-11T	41.6165	141.9924	55.27	4.6	mb	43	123	0.96	1.09	us	usb000g4ni	2013-04-111
12	2013-04-111		51.6758	10.07		mb		64	10.83			usb000g4mt	2013-04-11T
13	2013-04-10T	18.854	97.5096	8.27	4.7	mb	29	75	0.63	0.6		usb000g4i5	2013-04-111
14	2013-04-10T	20.8187	122.1203	4.2	5.8	Mww	115	31	2.21	1.28		usb000g4ca	2013-04-10T
15	2013-04-10T	2.6017	127.2174	66.02	5	mb	60	105	1.82	1.03	us	usb000g4br	2013-04-10T
16	2013-04-10T	15.5366	-87.228			mb		37	1.471	0.85		usb000g4a2	2013-04-11T
17	2013-04-10T	-10.7302	-75.2622	99.62		mb		72	1.99	0.77		usb000g43v	2013-04-101
18	2013-04-10T	-17.7569	167.7868	10		mb	20	160	3.71171	0.87		us2013nvap	2013-04-10T
19	2013-04-10T	28.5135	51.5523	9.93				94	10.877			usb000g3y3	2013-04-101
20	2013-04-10T		51.738	9.87		mb	76	58	10.87	1.03		usb000g3ts	2013-04-10T
21	2013-04-10T		\$1.7514	10.06		mb		75	10.974	0.95		usb000g3t2	2013-04-101
22	2013-04-10T		142.0723	27.79		mb	40	132	3.23	1.17		usb000g3qe	2013-04-10T
23	2013-04-10T		\$1.6075	10.02		mb	76	25	10.91	1.14		usb000g3p7	2013-04-101
24	2013-04-10T		139.0662	\$5.19	4.8	mb	33	61	6.81	1.6		usb000g3ns	2013-04-10T
25	2013-04-10T		51.604	10		mb		139	10.883	0.83		usb000g3nn	2013-04-10T
26	2013-04-10T		-79.5666	103.34		mb	35	113	2.56			usb000g3ng	2013-04-11T
27	2013-04-09T	-22.7541	69.1376	10.2		mb	17	110	11.11	0.55		usb000g3ls	2013-04-09T
28	2013-04-09T	28.2759	\$1.6754	9.88		mb		86	11.034	0.68		usb000g3ft	2013-04-09T
29	2013-04-09T	5.6129	93.3101	31.21		mb		139	3.641	0.61		usb000g3fe	2013-04-09T
	2013-04-09T	28.4201	51.6408	19.93	4.6	mb	94	94	10.92	0.72	us	usb000g3dn	2013-04-09T

(USGS Volcano Hazards Program, May 2018)

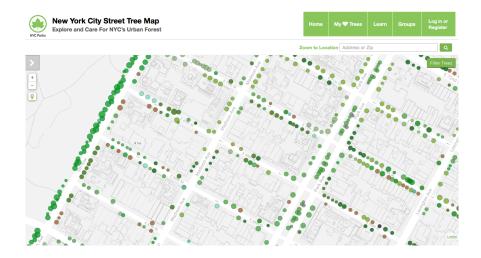
Demo turtles solution.

CSci 127 (Hunter)

Lecture 13

8 May 2018 26 / 44

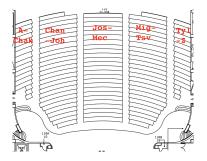
Design Question: Street Trees



CSci 127 (Hunter)

Lecture 13

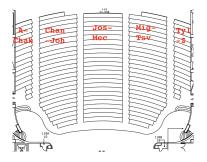
8 May 2018 27 / 44



• Practice exam: the same format as the final (except 1, not full 2 hours).

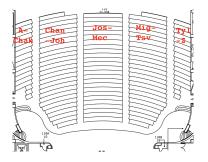
- $\exists \rightarrow$ 990

・ロト ・回ト ・ヨト



Practice exam: the same format as the final (except 1, not full 2 hours).Bring ID & 1 page of notes (will check IDs during exam).

Image: A math a math



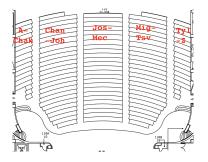
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.

CSci 127 (Hunter)

Lecture 13

8 May 2018 28 / 44

Image: A math a math



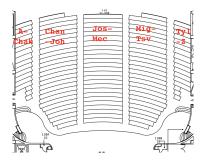
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.
 - Seating: every other seat, every other row.

CSci 127 (Hunter)

Lecture 13

8 May 2018 28 / 44

Image: A math a math

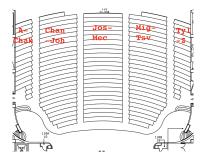


- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.
 - Seating: every other seat, every other row.
 - Sign in as you enter (clipboards for each section of the alphabet).

CSci 127 (Hunter)

8 May 2018 28 / 44

< □ > < 同 > < 三 >



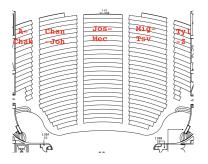
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.
 - Seating: every other seat, every other row.
 - ► Sign in as you enter (clipboards for each section of the alphabet).
 - Sign out as you leave.

CSci 127 (Hunter)

Lecture 13

8 May 2018 28 / 44

Image: A match a ma

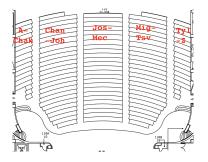


- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.
 - Seating: every other seat, every other row.
 - ► Sign in as you enter (clipboards for each section of the alphabet).
 - Sign out as you leave.
- Lecture slip for next week: signing out at end of mock exam.

CSci 127 (Hunter)

8 May 2018 28 / 44

イロト イロト イヨト



- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.
 - Seating: every other seat, every other row.
 - ► Sign in as you enter (clipboards for each section of the alphabet).
 - Sign out as you leave.
- Lecture slip for next week: signing out at end of mock exam.
- Answer key will be available on webpage after lecture.

CSci 127 (Hunter)

Lecture 13

8 May 2018 28 / 44

#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

Sar

#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 (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>
  3
  return 0;
```

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

• Definite Loops in Python & C++

8 May 2018 29 / 44

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 (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>
  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++

8 May 2018 29 / 44

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 (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>
  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)

イロト 不得 トイヨト イヨト ヨー のくや

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

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 - ����

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

900

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

Sac

- 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!!")
```

イロト 不良 トイヨト イヨト ヨー のくや

• 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.

Sac

<ロト <回ト < 回ト < 回ト

 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:

Sac

イロト イポト イラト イラト 一支

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.

Write C++ code that repeatedly prompts until an odd number is entered.
 CSci 127 (Hunter)
 Lecture 13
 KMay 2018
 KMay 2018
 KMay 2018

```
• 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.

200

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

Sac

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