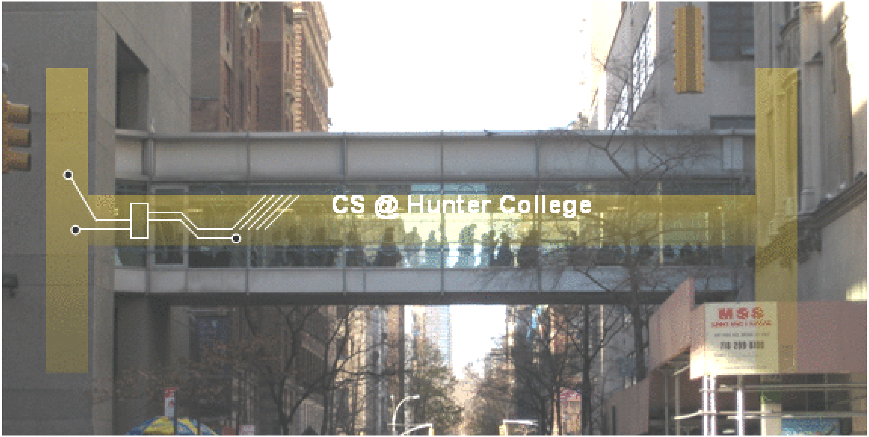
CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?
  We're part of a pilot program for prospective students. Last visit will be next week.

Frequently Asked Questions

From lecture slips & recitation sections.

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

Frequently Asked Questions

From lecture slips & recitation sections.

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

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?
  We're part of a pilot program for prospective students. Last visit will be next week.
- 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).

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?
  We're part of a pilot program for prospective students. Last visit will be next week.
- 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%.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?
  We're part of a pilot program for prospective students. Last visit will be next week.
- 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%)* ⇒ *70%.*
  - $\frac{1}{3}$ *programs (10%) & 100% on final (70%)* ⇒ *80%.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?
  We're part of a pilot program for prospective students. Last visit will be next week.
- 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%.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?
  We're part of a pilot program for prospective students. Last visit will be next week.
- 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%)* ⇒ *70%.*
  - $\frac{1}{3}$ *programs (10%) & 100% on final (70%)* ⇒ *80%.*
  - *All programs (30%) & 60% on final (42%)* ⇒ *72%.*

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

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?
  We're part of a pilot program for prospective students. Last visit will be next week.
- 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?

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?
  We're part of a pilot program for prospective students. Last visit will be next week.
- 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%) ⇒ 70%.*
  - $\frac{1}{3}$ *programs (10%) & 100% on final (70%) ⇒ 80%.*
  - *All programs (30%) & 60% on final (42%) ⇒ 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++, MWTh 12:10-1pm + section) & CSci 150 (Discrete Structures, MTh 1:10-2:25pm + section)*

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?
  We're part of a pilot program for prospective students. Last visit will be next week.
- 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%) ⇒ 70%.*
    - $\frac{1}{3}$ *programs (10%) & 100% on final (70%) ⇒ 80%.*
    - *All programs (30%) & 60% on final (42%) ⇒ 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++, MWTh 12:10-1pm + section) & CSci 150 (Discrete Structures, MTh 1:10-2:25pm + section)*
    - *Minors: CSci 133 (More Python: multiple times) & CSci 232 (Databases, multiple times)*

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?
  We're part of a pilot program for prospective students. Last visit will be next week.
- 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%) ⇒ 70%.*
  - $\frac{1}{3}$ *programs (10%) & 100% on final (70%) ⇒ 80%.*
  - *All programs (30%) & 60% on final (42%) ⇒ 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++, MWTh 12:10-1pm + 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...

Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?
  We're part of a pilot program for prospective students. Last visit will be next week.
- 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++, MWTh 12:10-1pm + 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...
  It's a practice exam that we're holding next lecture.
  More details at the end of lecture.

Announcements

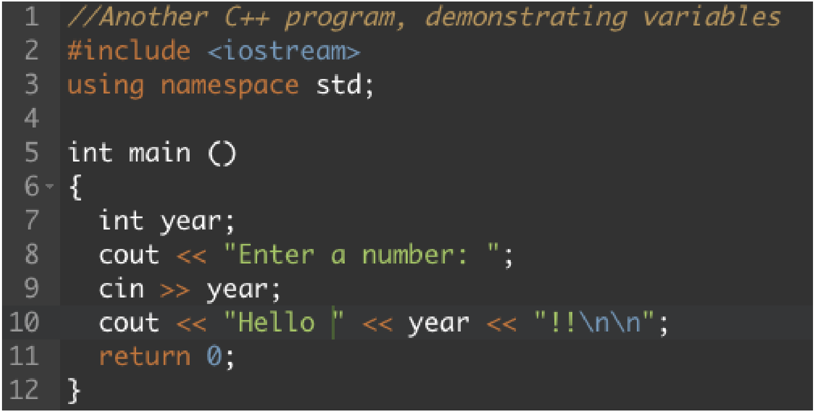
- Two handouts today:
  - Lecture slip, and
  - Final exam plans (pink).

Today's Topics

- Introducing C++: Basic Format & Variables
- I/O and Definite Loops in C++
- More Info on the Final Exam

In Pairs or Triples:

- Write a complete **Python program** that converts kilograms to pounds.
- *Predict what the C++ code will do:*



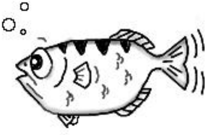
```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

Python Tutor

- Write a complete **Python program** that converts kilograms to pounds.

(Write from scratch in `pythonTutor`.)

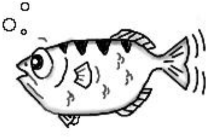
Side Note: gdb



gdb.org

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

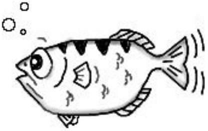
Side Note: gdb



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.

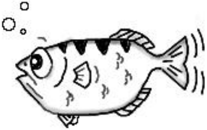
Side Note: gdb



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.

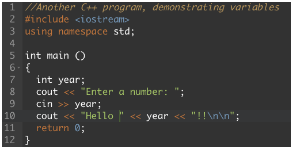
Side Note: gdb



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).

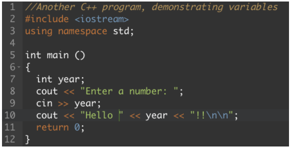
onlinegdb demo



```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

(Demo with `onlinegdb`)

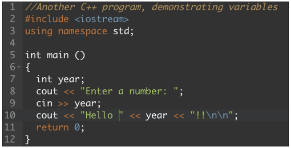
Introduction to C++



```cpp
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- C++ is a popular programming language that extends C.

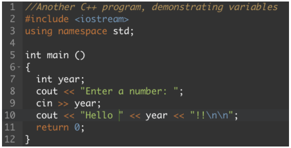
Introduction to C++



```cpp
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- C++ is a popular programming language that extends C.
- Fast, efficient, and powerful.

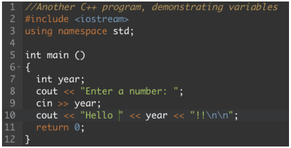
Introduction to C++



```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- C++ is a popular programming language that extends C.
- Fast, efficient, and powerful.
- Used for systems programming (and future courses!).

Introduction to C++

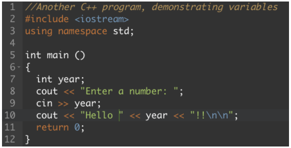


```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- C++ is a popular programming language that extends C.
- Fast, efficient, and powerful.
- Used for systems programming (and future courses!).
- Today, we'll introduce the basic structure and simple input/output (I/O) in C/C++.

Introduction to C++

- Programs are organized in functions.

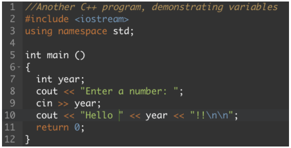


```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

Introduction to C++

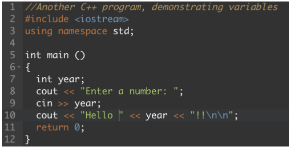
- Programs are organized in functions.

Example:



```cpp
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

Introduction to C++



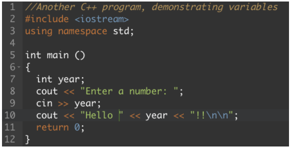
```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- Programs are organized in functions.

Example:

`int main()`

Introduction to C++



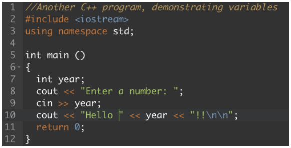
```cpp
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- Programs are organized in functions.

Example:

```
int main()
{
```

Introduction to C++



```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

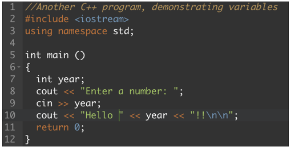
- Programs are organized in functions.

  Example:

```
int main()
{
     cout << "Hello world!";
     return(0);
}
```

Introduction to C++

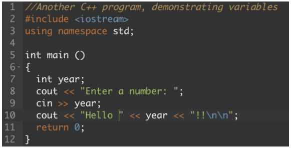
- Programs are organized in functions.



```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

Introduction to C++

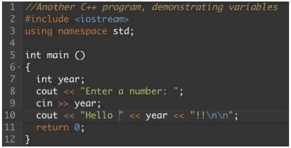
- Programs are organized in functions.
- Variables must be **declared**:



```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

Introduction to C++

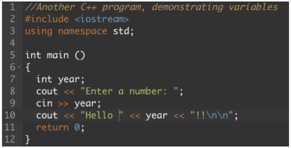
- Programs are organized in functions.
- Variables must be **declared**:
  `int num;`



```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

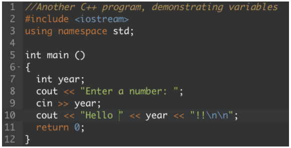
Introduction to C++

- Programs are organized in functions.
- Variables must be **declared**:
  `int num;`
- Many types available:
  `int, float, char, ...`



```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

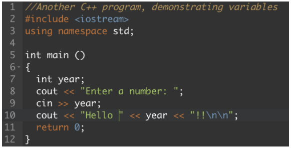
Introduction to C++



```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- Programs are organized in functions.
- Variables must be **declared**:
  `int num;`
- Many types available:
  `int, float, char, ...`
- Semicolons separate commands:

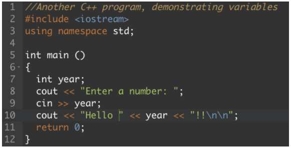
Introduction to C++



```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- Programs are organized in functions.
- Variables must be **declared**:
  `int num;`
- Many types available:
  `int, float, char, ...`
- Semicolons separate commands:
  `num = 5; more = 2*num;`

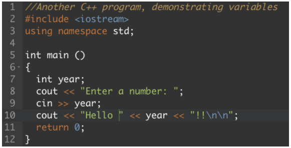
Introduction to C++



```cpp
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- Programs are organized in functions.
- Variables must be **declared**:
  `int num;`
- Many types available:
  `int, float, char, ...`
- Semicolons separate commands:
  `num = 5; more = 2*num;`
- To print, we'll use `cout <<`:

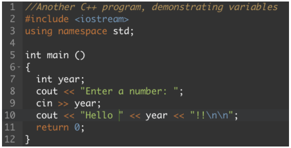
Introduction to C++



```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- Programs are organized in functions.
- Variables must be **declared**:
  `int num;`
- Many types available:
  `int, float, char, ...`
- Semicolons separate commands:
  `num = 5; more = 2*num;`
- To print, we'll use `cout <<`:
  `cout << "Hello!!";`

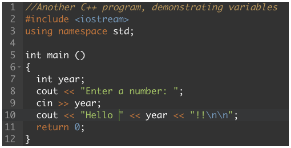
Introduction to C++



```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- Programs are organized in functions.
- Variables must be **declared**:
  `int num;`
- Many types available:
  `int, float, char, ...`
- Semicolons separate commands:
  `num = 5; more = 2*num;`
- To print, we'll use `cout <<`:
  `cout << "Hello!!";`
- To get input, we'll use `cin >>`:

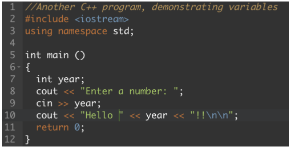
Introduction to C++



```cpp
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- Programs are organized in functions.
- Variables must be **declared**:
  `int num;`
- Many types available:
  `int, float, char, ...`
- Semicolons separate commands:
  `num = 5; more = 2*num;`
- To print, we'll use `cout <<`:
  `cout << "Hello!!";`
- To get input, we'll use `cin >>`:
  `cin >> num;`

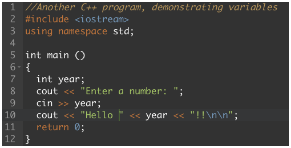
Introduction to C++



```cpp
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- Programs are organized in functions.
- Variables must be **declared**:
  `int num;`
- Many types available:
  `int, float, char, ...`
- Semicolons separate commands:
  `num = 5; more = 2*num;`
- To print, we'll use `cout <<`:
  `cout << "Hello!!";`
- To get input, we'll use `cin >>`:
  `cin >> num;`
- To use those I/O functions, we put at the top of the program:

Introduction to C++

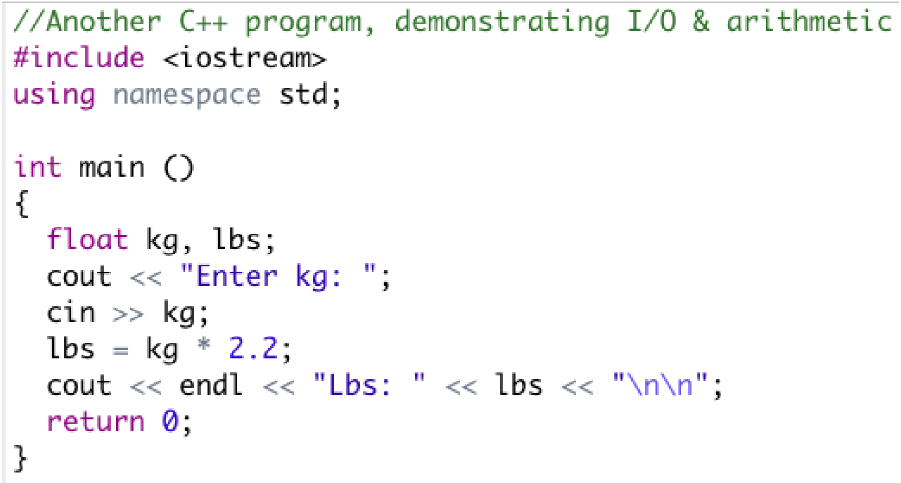


```
//Another C++ program, demonstrating variables
#include <iostream>
using namespace std;

int main ()
{
  int year;
  cout << "Enter a number: ";
  cin >> year;
  cout << "Hello " << year << "!!\n\n";
  return 0;
}
```

- Programs are organized in functions.
- Variables must be **declared**:
  `int num;`
- Many types available:
  `int, float, char, ...`
- Semicolons separate commands:
  `num = 5; more = 2*num;`
- To print, we'll use `cout <<`:
  `cout << "Hello!!";`
- To get input, we'll use `cin >>`:
  `cin >> num;`
- To use those I/O functions, we put at the top of the program:
  `#include <iostream>`
  `using namespace std;`

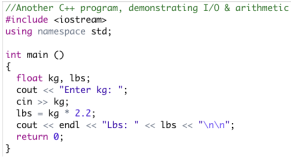
In Pairs or Triples:

Predict what the following pieces of code will do:



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

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

C++ Demo



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

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

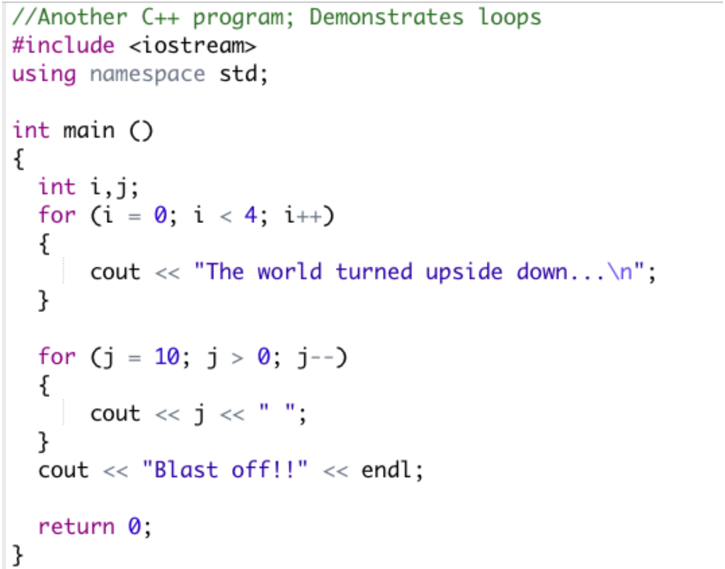
(Demo with `onlinegdb`)

Today's Topics

- Introducing C++: Basic Format & Variables
- **I/O and Definite Loops in C++**
- More Info on the Final Exam

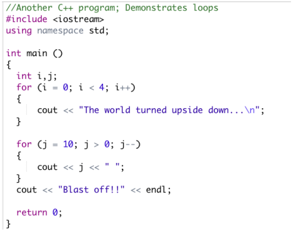
In Pairs or Triples:

Predict what the following pieces of code will do:



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

int main ()
{
  int i,j;
  for (i = 0; i < 4; i++)
  {
      cout << "The world turned upside down...\n";
  }

  for (j = 10; j > 0; j--)
  {
      cout << j << " ";
  }
  cout << "Blast off!!" << endl;

  return 0;
}
```

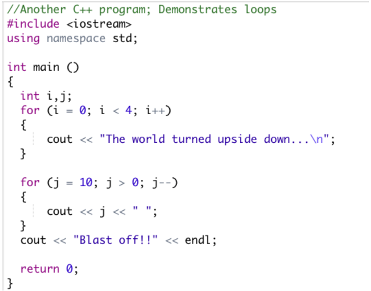
C++ Demo



```
//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";
  }

  for (j = 10; j > 0; j--)
  {
      cout << j << " ";
  }
  cout << "Blast off!!" << endl;

  return 0;
}
```

(Demo with `onlinegdb`)

Definite loops



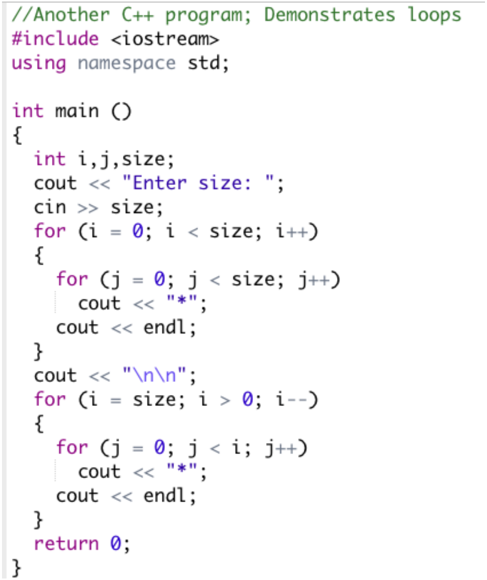
```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;

int main ()
{
  int i,j;
  for (i = 0; i < 4; i++)
  {
      cout << "The world turned upside down...\n";
  }

  for (j = 10; j > 0; j--)
  {
      cout << j << " ";
  }
  cout << "Blast off!!" << endl;

  return 0;
}
```

General format:

```
for ( initialization ; test ; updateAction )
{

        command1;
        command2;
        command3;
        ...
}
```

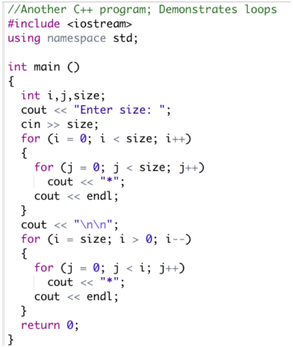
In Pairs or Triples:

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: ";
  cin >> size;
  for (i = 0; i < size; i++)
  {
    for (j = 0; j < size; j++)
      cout << "*";
    cout << endl;
  }
  cout << "\n\n";
  for (i = size; i > 0; i--)
  {
    for (j = 0; j < i; j++)
      cout << "*";
    cout << endl;
  }
  return 0;
}
```

C++ Demo

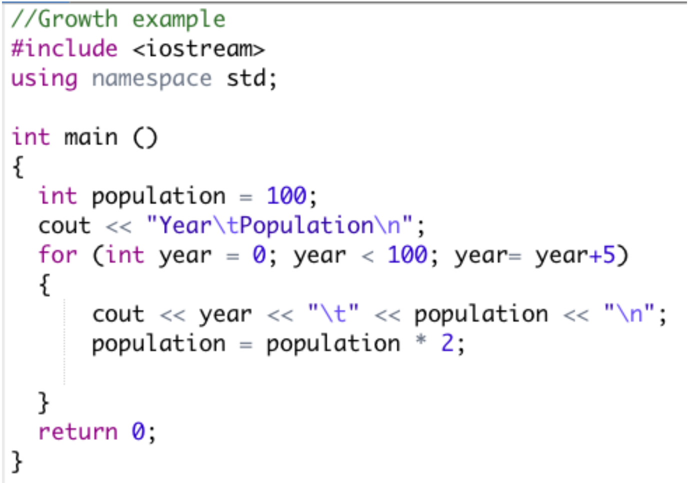


```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;

int main ()
{
  int i,j,size;
  cout << "Enter size: ";
  cin >> size;
  for (i = 0; i < size; i++)
  {
    for (j = 0; j < size; j++)
      cout << "*";
    cout << endl;
  }
  cout << "\n\n";
  for (i = size; i > 0; i--)
  {
    for (j = 0; j < i; j++)
      cout << "*";
    cout << endl;
  }
  return 0;
}
```

(Demo with C++)

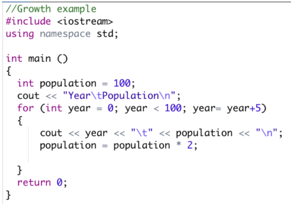
In Pairs or Triples:

Predict what the following pieces of code will do:



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

int main ()
{
  int population = 100;
  cout << "Year\tPopulation\n";
  for (int year = 0; year < 100; year= year+5)
  {
      cout << year << "\t" << population << "\n";
      population = population * 2;

  }
  return 0;
}
```

C++ Demo

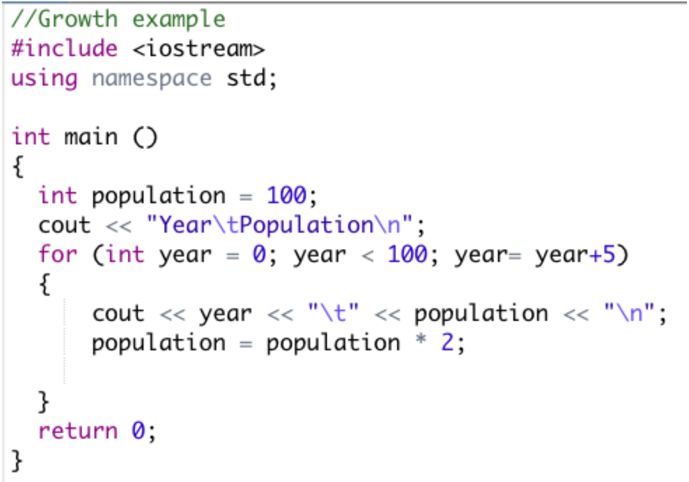


```
//Growth example
#include <iostream>
using namespace std;

int main ()
{
  int population = 100;
  cout << "Year\tPopulation\n";
  for (int year = 0; year < 100; year= year+5)
  {
      cout << year << "\t" << population << "\n";
      population = population * 2;

  }
  return 0;
}
```

(Demo with C++)

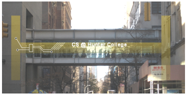
Lecture Slips

In pairs or triples: **translate** the C++ program into Python:



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

int main ()
{
  int population = 100;
  cout << "Year\tPopulation\n";
  for (int year = 0; year < 100; year= year+5)
  {
      cout << year << "\t" << population << "\n";
      population = population * 2;

  }
  return 0;
}
```

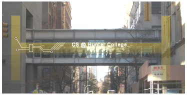
Recap: C++



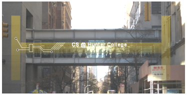
- On lecture slip, write down a topic you wish we had spent more time (and why).

Recap: C++

- On lecture slip, write down a topic you wish we had spent more time (and why).
- C++ is a popular programming language that extends C.

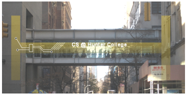


Recap: C++



- On lecture slip, write down a topic you wish we had spent more time (and why).
- C++ is a popular programming language that extends C.
- Input/Output (I/O):
  - `cin` >>
  - `cout` <<

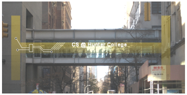
Recap: C++



- On lecture slip, write down a topic you wish we had spent more time (and why).
- C++ is a popular programming language that extends C.
- Input/Output (I/O):
  - `cin >>`
  - `cout <<`
- Definite loops:

```
for (i = 0; i < 10; i++) {
     ...
}
```

Recap: C++



- On lecture slip, write down a topic you wish we had spent more time (and why).
- C++ is a popular programming language that extends C.
- Input/Output (I/O):
  - `cin >>`
  - `cout <<`
- Definite loops:

```
for (i = 0; i < 10; i++) {
     ...
}
```

- Pass your lecture slip to the aisles for UTA's to collect.

Today's Topics

- Introducing C++: Basic Format & Variables
- I/O and Definite Loops in C++
- **More Info on the Final Exam**

Final Exam: When

- The final exam is **Wednesday, 19 December, 9am-11am**, Assembly Hall (118 HN).

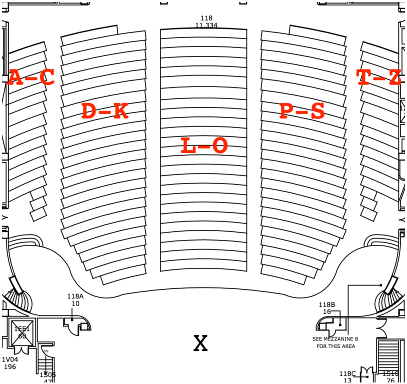
Final Exam: When

- The final exam is **Wednesday, 19 December, 9am-11am**, Assembly Hall (118 HN).
- If you have a conflict, the alternative time is: Thursday, 12 December, 1:45pm-3:45pm, 1001E HN.

Final Exam: When

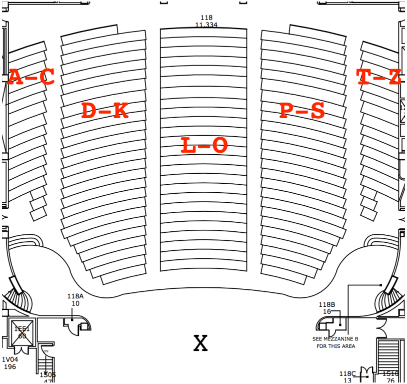
- The final exam is **Wednesday, 19 December, 9am-11am**, Assembly Hall (118 HN).
- If you have a conflict, the alternative time is:
  Thursday, 12 December, 1:45pm-3:45pm, 1001E HN.
- If you have accommodations via the Accessibility Office, we will send the exam to their testing center.
  (Must complete by end of day, Tuesday, 18 December.)

Final Exam: Logistics



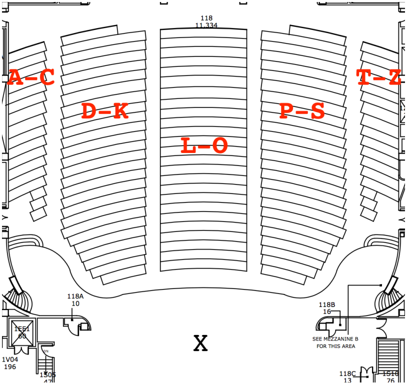
- Bring ID, note sheet, pencils or pens.

Final Exam: Logistics



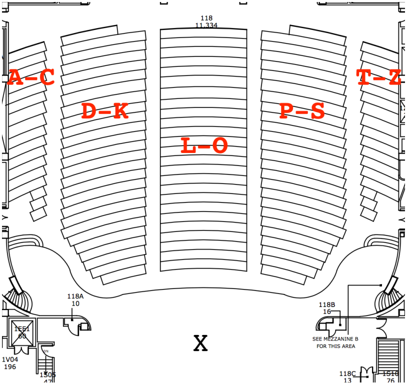
- Bring ID, note sheet, pencils or pens.
- Seating is by last name– sign in as you enter.

Final Exam: Logistics



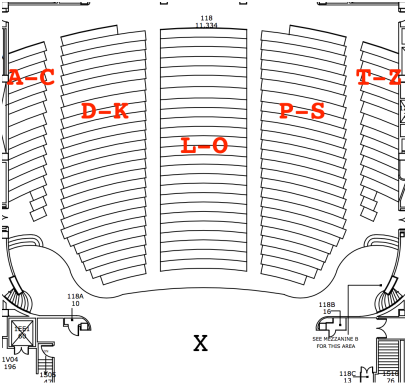
- Bring ID, note sheet, pencils or pens.
- Seating is by last name– sign in as you enter.
- Sign out when you turn in your exam.

Final Exam: Logistics



- Bring ID, note sheet, pencils or pens.
- Seating is by last name– sign in as you enter.
- Sign out when you turn in your exam.
- Cannot leave during the first 45 minutes of the exam.

Final Exam: Logistics



- Bring ID, note sheet, pencils or pens.
- Seating is by last name– sign in as you enter.
- Sign out when you turn in your exam.
- Cannot leave during the first 45 minutes of the exam.
- Cannot start the exam after students start leaving.

Final Exam: Format

- The exam is 2 hours long.

Final Exam: Format

- The exam is 2 hours long.
- There are 4 different versions to discourage copying.

Final Exam: Format

- The exam is 2 hours long.
- There are 4 different versions to discourage copying.
- It is on paper. No use of computers, phones, etc. allowed.

Final Exam: Format

- The exam is 2 hours long.
- There are 4 different versions to discourage copying.
- It is on paper. No use of computers, phones, etc. allowed.
- You may have 1 piece of **8.5" x 11"** piece of paper.

Final Exam: Format

- The exam is 2 hours long.
- There are 4 different versions to discourage copying.
- It is on paper. No use of computers, phones, etc. allowed.
- You may have 1 piece of **8.5" x 11"** piece of paper.
    - With notes, examples, programs: what will help you on the exam.

Final Exam: Format

- The exam is 2 hours long.
- There are 4 different versions to discourage copying.
- It is on paper. No use of computers, phones, etc. allowed.
- You may have 1 piece of **8.5” x 11”** piece of paper.
  - With notes, examples, programs: what will help you on the exam.
  - No origami– it’s distracting to others taking the exam.

Final Exam: Format

- The exam is 2 hours long.
- There are 4 different versions to discourage copying.
- It is on paper. No use of computers, phones, etc. allowed.
- You may have 1 piece of **8.5" x 11"** piece of paper.
  - With notes, examples, programs: what will help you on the exam.
  - No origami– it's distracting to others taking the exam.
  - Best if you design/write yours since excellent way to study.

Final Exam: Format

- The exam is 2 hours long.
- There are 4 different versions to discourage copying.
- It is on paper. No use of computers, phones, etc. allowed.
- You may have 1 piece of **8.5" x 11"** piece of paper.
    - With notes, examples, programs: what will help you on the exam.
    - No origami– it's distracting to others taking the exam.
    - Best if you design/write yours since excellent way to study.
- The exam format:

Final Exam: Format

- The exam is 2 hours long.
- There are 4 different versions to discourage copying.
- It is on paper. No use of computers, phones, etc. allowed.
- You may have 1 piece of **8.5" x 11"** piece of paper.
  - With notes, examples, programs: what will help you on the exam.
  - No origami– it's distracting to others taking the exam.
  - Best if you design/write yours since excellent way to study.
- The exam format:
  - Printed on both sides of the paper (thanks to new 2-sided scanners!).

Final Exam: Format

- The exam is 2 hours long.
- There are 4 different versions to discourage copying.
- It is on paper. No use of computers, phones, etc. allowed.
- You may have 1 piece of **8.5" x 11"** piece of paper.
  - With notes, examples, programs: what will help you on the exam.
  - No origami– it's distracting to others taking the exam.
  - Best if you design/write yours since excellent way to study.
- The exam format:
  - Printed on both sides of the paper (thanks to new 2-sided scanners!).
  - 10 questions, each worth 10 points.

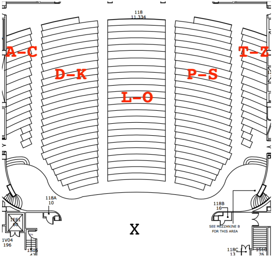
Final Exam: Format

- The exam is 2 hours long.
- There are 4 different versions to discourage copying.
- It is on paper. No use of computers, phones, etc. allowed.
- You may have 1 piece of **8.5" x 11"** piece of paper.
  - With notes, examples, programs: what will help you on the exam.
  - No origami– it's distracting to others taking the exam.
  - Best if you design/write yours since excellent way to study.
- The exam format:
  - Printed on both sides of the paper (thanks to new 2-sided scanners!).
  - 10 questions, each worth 10 points.
  - Style of questions: what does the code do? short answer, write functions, top down design, & write complete programs.

Final Exam: Format

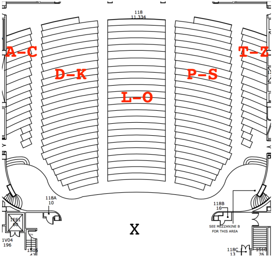
- The exam is 2 hours long.
- There are 4 different versions to discourage copying.
- It is on paper. No use of computers, phones, etc. allowed.
- You may have 1 piece of **8.5" x 11"** piece of paper.
  - With notes, examples, programs: what will help you on the exam.
  - No origami– it's distracting to others taking the exam.
  - Best if you design/write yours since excellent way to study.
- The exam format:
  - Printed on both sides of the paper (thanks to new 2-sided scanners!).
  - 10 questions, each worth 10 points.
  - Style of questions: what does the code do? short answer, write functions, top down design, & write complete programs.
- Past exams available on webpage (includes answer keys).

Mock Final



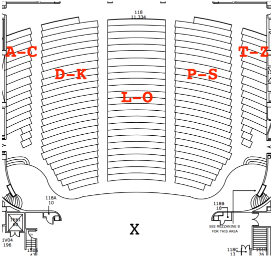
- Given in lecture on 12 December.

Mock Final



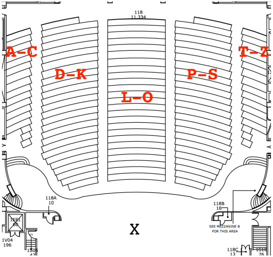
- Given in lecture on 12 December.
- Practice exam: the same format as the final (except 1, not full 2 hours).

Mock Final



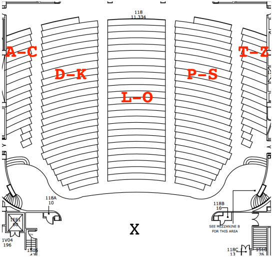
- Given in lecture on 12 December.
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).

Mock Final



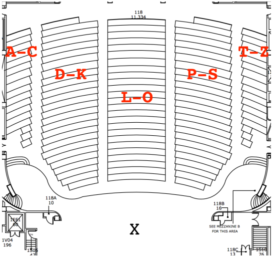
- Given in lecture on 12 December.
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-C, D-J, J-M, N-S, and T-Z.

Mock Final



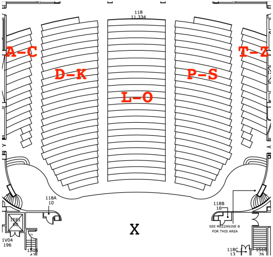
- Given in lecture on 12 December.
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-C, D-J, J-M, N-S, and T-Z.
  - Seating: every other seat, every other row.

Mock Final



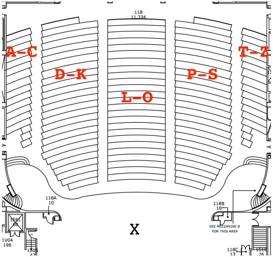
- Given in lecture on 12 December.
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-C, D-J, J-M, N-S, and T-Z.
  - Seating: every other seat, every other row.
  - Sign in as you enter (clipboards for each section of the alphabet).

Mock Final



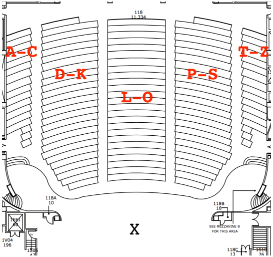
- Given in lecture on 12 December.
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-C, D-J, J-M, N-S, and T-Z.
  - Seating: every other seat, every other row.
  - Sign in as you enter (clipboards for each section of the alphabet).
  - Sign out as you leave (cannot leave in first 45 minutes).

Mock Final



- Given in lecture on 12 December.
- 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-C, D-J, J-M, N-S, and T-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 (cannot leave in first 45 minutes).
- Lecture slip for that week: signing out at end of mock exam.

Mock Final



- Given in lecture on 12 December.
- 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-C, D-J, J-M, N-S, and T-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 (cannot leave in first 45 minutes).
- Lecture slip for that week: signing out at end of mock exam.
- Answer key will be available on webpage after lecture.

Practice Quiz & Final Questions

- Lightning rounds:

Practice Quiz & Final Questions

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

Practice Quiz & Final Questions

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

Practice Quiz & Final Questions

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

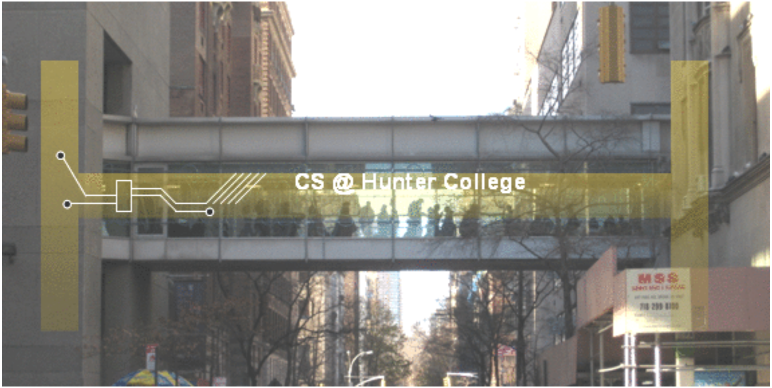
Practice Quiz & Final Questions

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

Practice Quiz & Final Questions

- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.
- Past exams are on the webpage (under Final Exam Information).
- We'll start with Spring 18, Version 1.

Writing Boards



- Return writing boards as you leave...