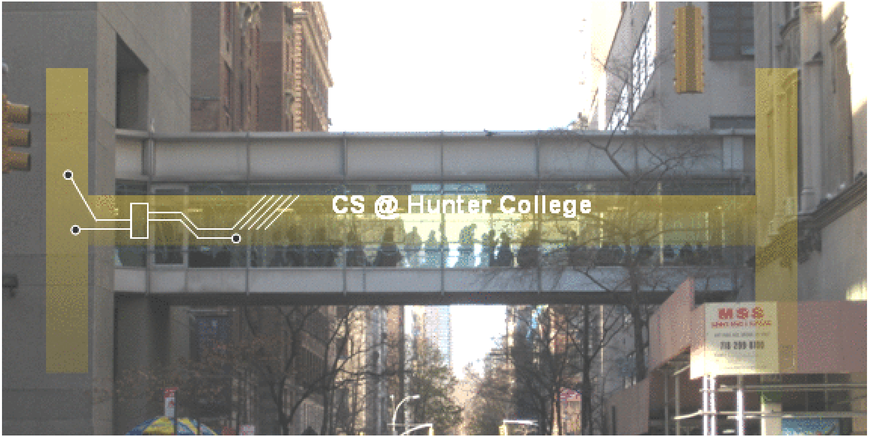
CSci 127: Introduction to Computer Science



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 is today.

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 is today.
- 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 is today.
- 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 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).

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 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%)* ⇒ *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 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%.*

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 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%.*

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 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%) ⇒ 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 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%) ⇒ 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?

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 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)*

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 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%) ⇒ 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++, 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)*

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 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%) ⇒ 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++, 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...

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 is today.

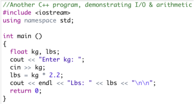
- I'm worried about my grade. Should I do Credit/NoCredit?
  Programs are worth 30% of the final grade.
  Lecture slips & in-class quizzes can only help you (up to 35% of your grade).
  - *No programs & 100% on final (adds 70%) ⇒ 70%.*
  - $\frac{1}{3}$ *programs (10%) & 100% on final (70%) ⇒ 80%.*
  - *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++, 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...
  It's a practice exam that we're holding next lecture.
  More details at end of lecture.

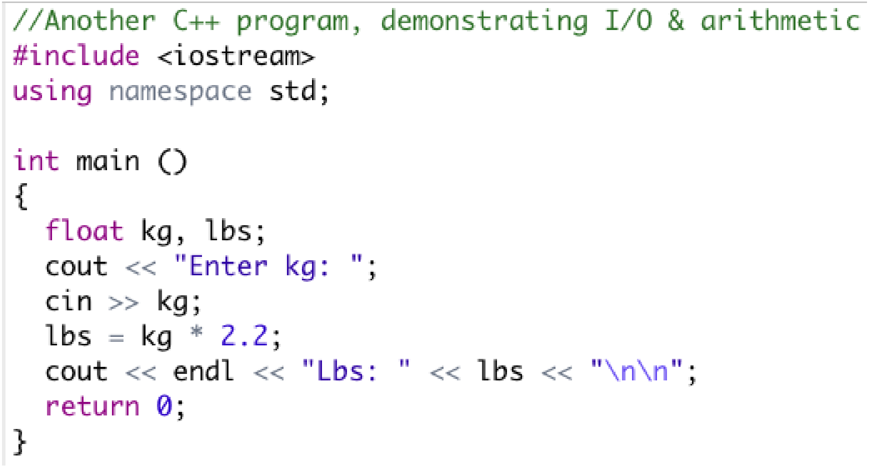
Today's Topics



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

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

- Recap: I/O & Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Review: Design & Final Questions

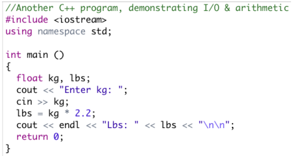
Introduction to C++



```cpp
//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;
}
```

Introduction to C++

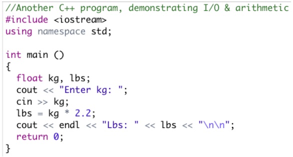
- Efficient for systems programming.



```
//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;
}
```

Introduction to C++

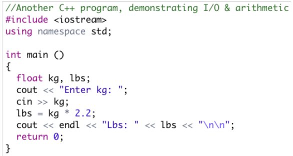
- Efficient for systems programming.
- Programs are organized in functions.



```
//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;
}
```

Introduction to C++

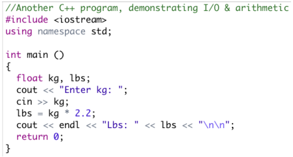
- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables:



```
//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;
}
```

Introduction to C++

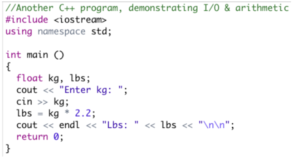
- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`



```
//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;
}
```

Introduction to C++

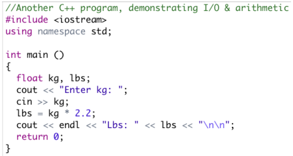
- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:



```
//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;
}
```

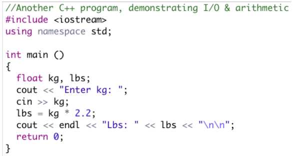
Introduction to C++

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



```
//Another C++ program, demonstrating I/O & arithmetic
#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;
}
```

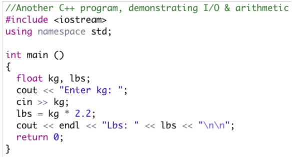
Introduction to C++



```
//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;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print:

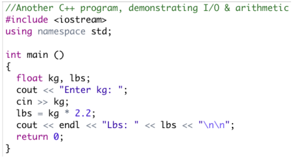
Introduction to C++



```
//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;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:
  `int, float, char, ...`
- To print: `cout << "Hello!!";`

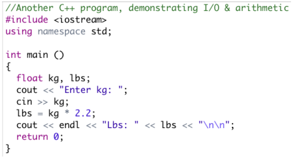
Introduction to C++



```
//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;
}
```

- 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:

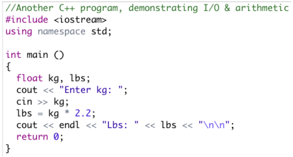
Introduction to C++



```
//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;
}
```

- 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;`

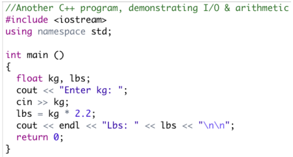
Introduction to C++



```
//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;
}
```

- 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:

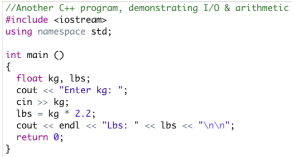
Introduction to C++



```
//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;
}
```

- 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;`

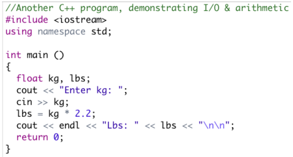
Introduction to C++



```cpp
//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;
}
```

- 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:

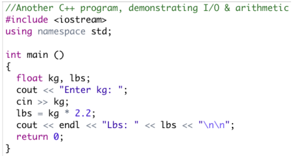
Introduction to C++



```
//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;
}
```

- 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++) {...}`

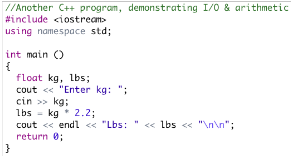
Introduction to C++



```
//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;
}
```

- 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++) {...}`
- Blocks of code uses '{' and '}'.

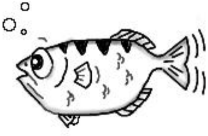
Introduction to C++



```
//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;
}
```

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

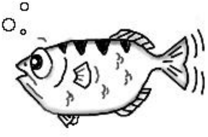
Side Note: `gdb`



gdb.org

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

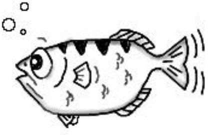
Side Note: gdb



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.

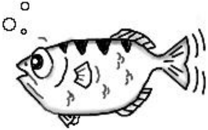
Side Note: gdb



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

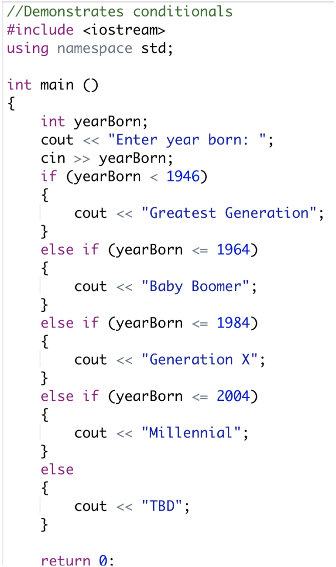


gdb.org

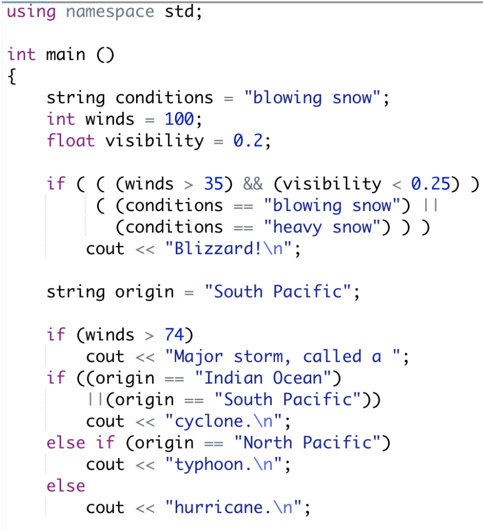
- Part of Richard Stallman's "GNU is Not Unix" (GNU) project.
- Written in 1986, gdb is the GNU debugger and based on dbx from the Berkeley Distribution of Unix.
- Lightweight, widely-available program that allows you to "step through" your code line-by-line.
- Available on the lab machines (via command-line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

In Pairs or Triples:

Predict what the following pieces of code will do:

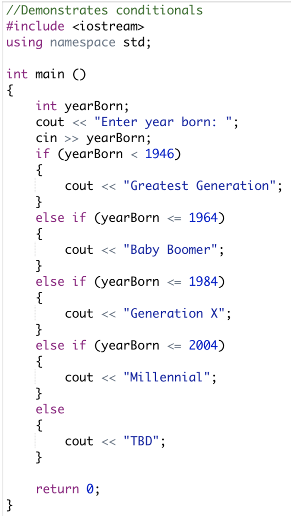


```
//Demonstrates conditionals
#include <iostream>
using namespace std;

int main ()
{
    int yearBorn;
    cout << "Enter year born: ";
    cin >> yearBorn;
    if (yearBorn < 1946)
    {
        cout << "Greatest Generation";
    }
    else if (yearBorn <= 1964)
    {
        cout << "Baby Boomer";
    }
    else if (yearBorn <= 1984)
    {
        cout << "Generation X";
    }
    else if (yearBorn <= 2004)
    {
        cout << "Millennial";
    }
    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";

    string origin = "South Pacific";

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

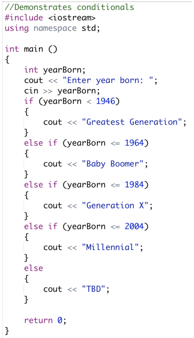
C++ Demo



```
//Demonstrates conditionals
#include <iostream>
using namespace std;

int main ()
{
    int yearBorn;
    cout << "Enter year born: ";
    cin >> yearBorn;
    if (yearBorn < 1946)
    {
        cout << "Greatest Generation";
    }
    else if (yearBorn <= 1964)
    {
        cout << "Baby Boomer";
    }
    else if (yearBorn <= 1984)
    {
        cout << "Generation X";
    }
    else if (yearBorn <= 2004)
    {
        cout << "Millennial";
    }
    else
    {
        cout << "TBD";
    }

    return 0;
}
```

(Demo with `onlinegdb`)

Conditionals



```
//Demonstrates conditionals
#include <iostream>
using namespace std;

int main ()
{
    int yearBorn;
    cout << "Enter year born: ";
    cin >> yearBorn;
    if (yearBorn < 1946)
    {
        cout << "Greatest Generation";
    }
    else if (yearBorn <= 1964)
    {
        cout << "Baby Boomer";
    }
    else if (yearBorn <= 1984)
    {
        cout << "Generation X";
    }
    else if (yearBorn <= 2004)
    {
        cout << "Millennial";
    }
    else
    {
        cout << "TBD";
    }

    return 0;
}
```

General format:

```
if ( logical expression )
{
      command1;
      ...
}
else if ( logical expression )
{
      command1;
      ...
}
else
{
      command1;
      ...
}
```

Logical Operators in C++

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

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 (`||`)

<table>
<tr><th>in1</th><th></th><th>in2</th><th><i>returns:</i></th></tr>
<tr><td>False</td><td>||</td><td>False</td><td>False</td></tr>
<tr><td>False</td><td>||</td><td>True</td><td>True</td></tr>
<tr><td>True</td><td>||</td><td>False</td><td>True</td></tr>
<tr><td>True</td><td>||</td><td>True</td><td>True</td></tr>
</table>

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 (||)

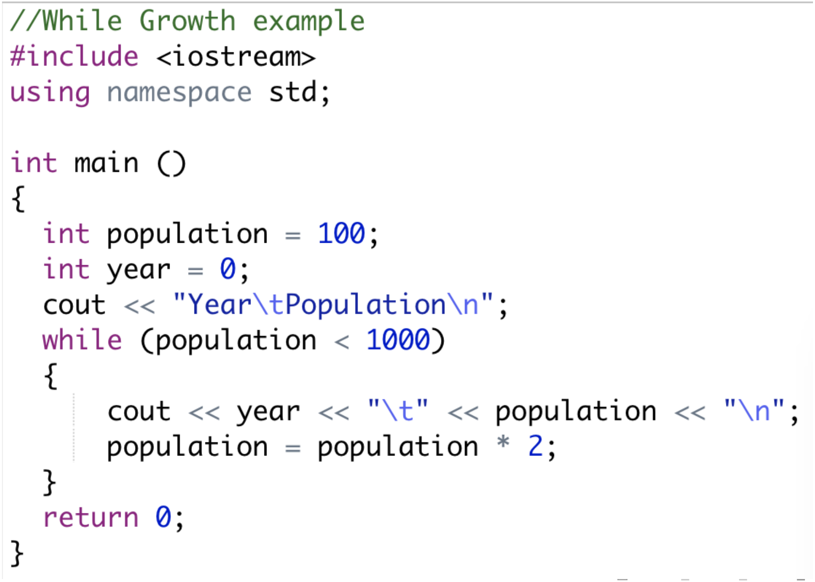
in1		in2	*returns:*
False	||	False	False
False	||	True	True
True	||	False	True
True	||	True	True

not (!)

	in1	*returns:*
!	False	True
!	True	False

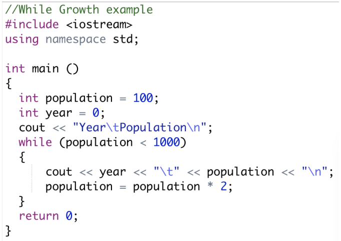
In Pairs or Triples:

Predict what the following pieces of code will do:



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

int main ()
{
  int population = 100;
  int year = 0;
  cout << "Year\tPopulation\n";
  while (population < 1000)
  {
      cout << year << "\t" << population << "\n";
      population = population * 2;
  }
  return 0;
}
```

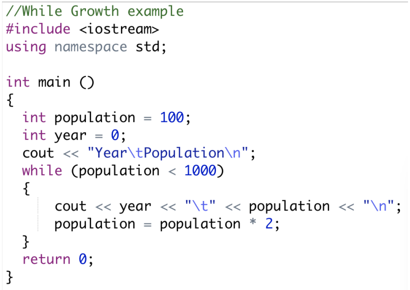
C++ Demo



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

int main ()
{
  int population = 100;
  int year = 0;
  cout << "Year\tPopulation\n";
  while (population < 1000)
  {
      cout << year << "\t" << population << "\n";
      population = population * 2;
  }
  return 0;
}
```

(Demo with `onlinegdb`)

Indefinite Loops: `while`



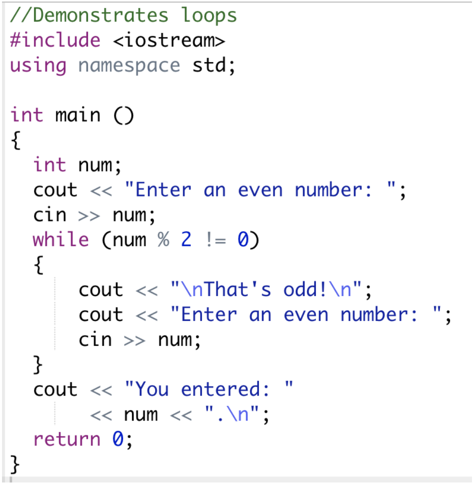
```
//While Growth example
#include <iostream>
using namespace std;

int main ()
{
  int population = 100;
  int year = 0;
  cout << "Year\tPopulation\n";
  while (population < 1000)
  {
      cout << year << "\t" << population << "\n";
      population = population * 2;
  }
  return 0;
}
```

General format:

```
while ( logical expression )
{

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

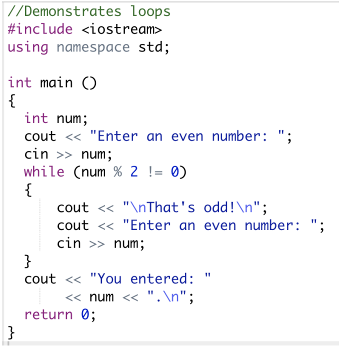
In Pairs or Triples:

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: ";
  cin >> num;
  while (num % 2 != 0)
  {
      cout << "\nThat's odd!\n";
      cout << "Enter an even number: ";
      cin >> num;
  }
  cout << "You entered: "
       << num << ".\n";
  return 0;
}
```

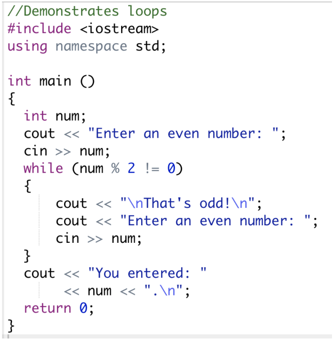
C++ Demo



```
//Demonstrates loops
#include <iostream>
using namespace std;

int main ()
{
  int num;
  cout << "Enter an even number: ";
  cin >> num;
  while (num % 2 != 0)
  {
      cout << "\nThat's odd!\n";
      cout << "Enter an even number: ";
      cin >> num;
  }
  cout << "You entered: "
       << num << ".\n";
  return 0;
}
```

(Demo with `onlinegdb`)

Indefinite Loops: `while`



```
//Demonstrates loops
#include <iostream>
using namespace std;

int main ()
{
  int num;
  cout << "Enter an even number: ";
  cin >> num;
  while (num % 2 != 0)
  {
      cout << "\nThat's odd!\n";
      cout << "Enter an even number: ";
      cin >> num;
  }
  cout << "You entered: "
       << num << ".\n";
  return 0;
}
```

General format:

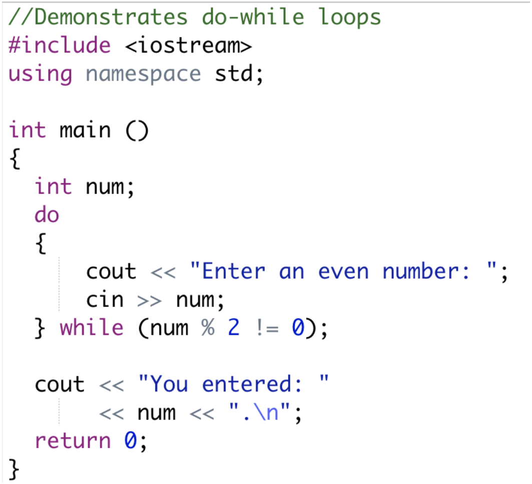
`while` (*logical expression*)
{

command1;
command2;
command3;
...

}

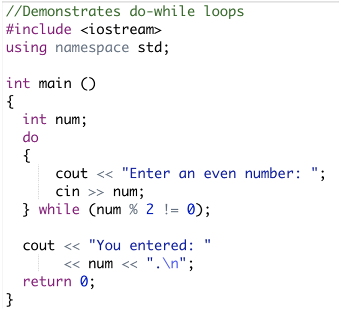
In Pairs or Triples:

Predict what the following pieces of code will do:



```
//Demonstrates do-while loops
#include <iostream>
using namespace std;

int main ()
{
  int num;
  do
  {
    cout << "Enter an even number: ";
    cin >> num;
  } while (num % 2 != 0);

  cout << "You entered: "
       << num << ".\n";
  return 0;
}
```

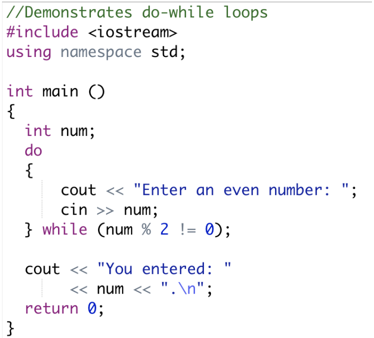
C++ Demo



```
//Demonstrates do-while loops
#include <iostream>
using namespace std;

int main ()
{
  int num;
  do
  {
      cout << "Enter an even number: ";
      cin >> num;
  } while (num % 2 != 0);

  cout << "You entered: "
       << num << ".\n";
  return 0;
}
```

(Demo with `onlinegdb`)

Indefinite Loops: `do-while`



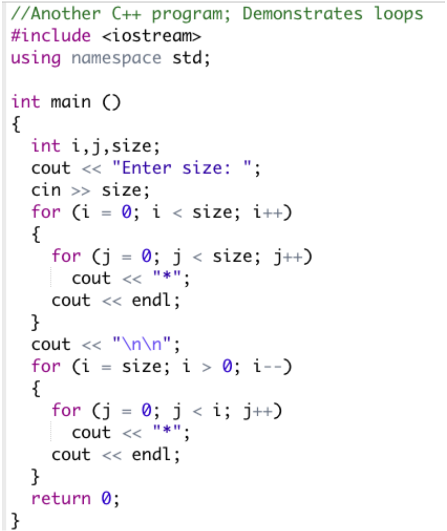
```
//Demonstrates do-while loops
#include <iostream>
using namespace std;

int main ()
{
  int num;
  do
  {
      cout << "Enter an even number: ";
      cin >> num;
  } while (num % 2 != 0);

  cout << "You entered: "
       << num << ".\n";
  return 0;
}
```

General format:

```
do
{
      command1;
      command2;
      command3;
      ...
}
while ( logical expression )
```

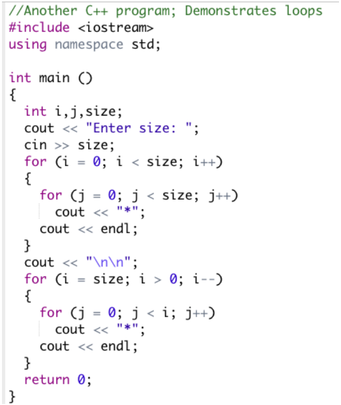
In Pairs or Triples:

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

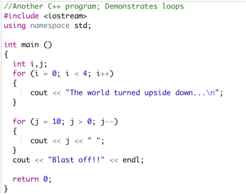


```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;

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 `onlinegdb`)

Recap: C++ Control Structures

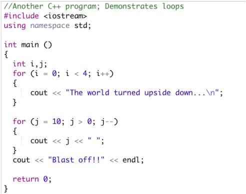
- I/O:



```
//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;
}
```

Recap: C++ Control Structures

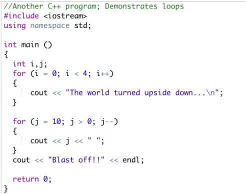
- I/O: `cin >> ...;`



```
//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;
}
```

Recap: C++ Control Structures

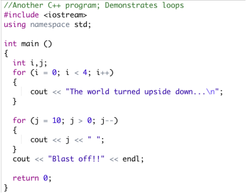
- I/O: `cin >> ...;` & `cout << ...;`



```
//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;
}
```

Recap: C++ Control Structures

- I/O: `cin >> ...;` & `cout << ...;`
- Definite loops:

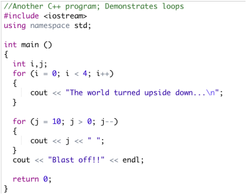


```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;

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;
}
```

Recap: C++ Control Structures

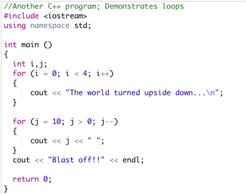
- I/O: `cin >> ...;` & `cout << ...;`
- Definite loops:

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



```
//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;
}
```

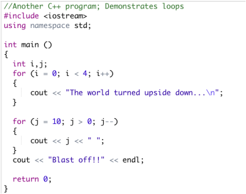
Recap: C++ Control Structures



```
//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;
}
```

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

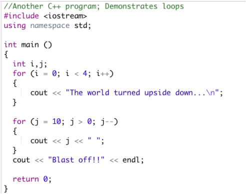
Recap: C++ Control Structures



```
//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;
}
```

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

Recap: C++ Control Structures



```
//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;
}
```

- I/O: `cin >> ...;` & `cout << ...;`
- Definite loops:

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

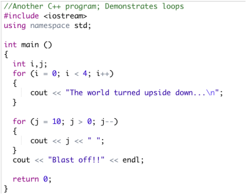
- Conditionals:

  `if` (*logical expression*)

  ```
  {
        ...
  }
  else
  {
        ...
  }
  ```

- Indefinite loops:

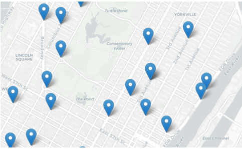
Recap: C++ Control Structures



```
//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;
}
```

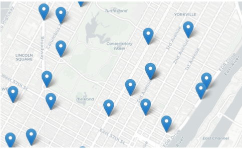
- I/O: `cin >> ...;` & `cout << ...;`
- Definite loops:
  ```
  for (i = 0; i < 10; i++)
  {
        ...
  }
  ```
- Conditionals:
  ```
  if (logical expression)
  {
        ...
  }
  else
  {
        ...
  }
  ```
- Indefinite loops:
  ```
  while (logical expression)
  {
        ...
  }
  ```

Design Questions & Lecture Slips



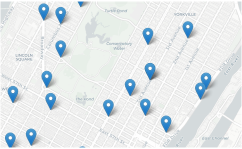
- Requests: More on design questions & the mock final.

Design Questions & Lecture Slips



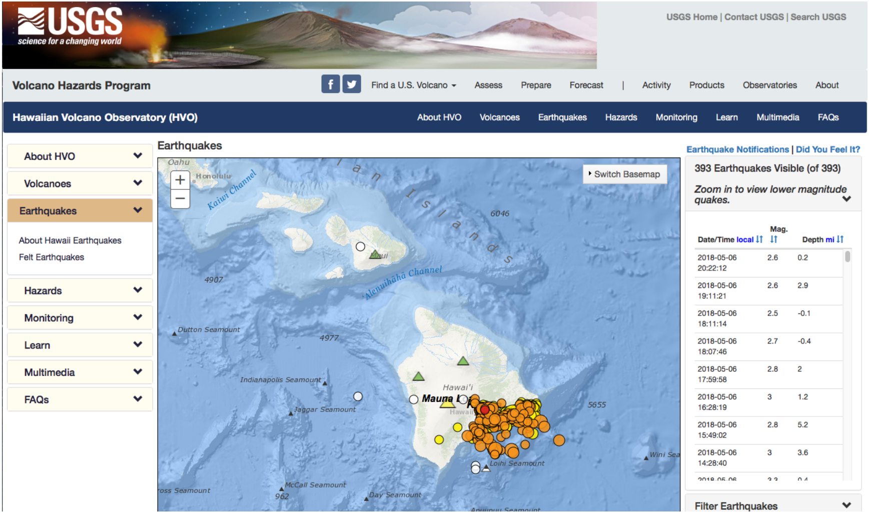
- Requests: More on design questions & the mock final.
- Extra lecture slip today for design questions.

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).

Design Question: Earthquakes



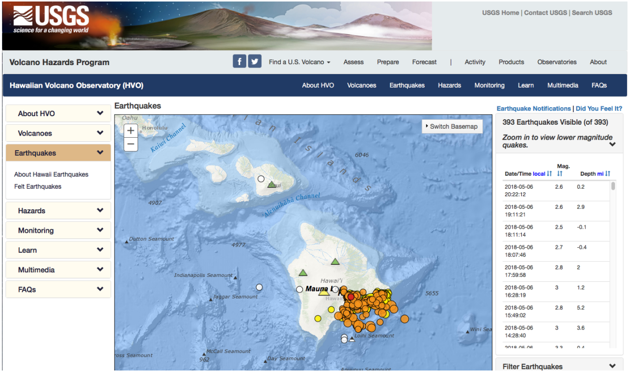
393 Earthquakes Visible (of 393)

Zoom in to view lower magnitude quakes.

Date/Time local	Mag.	Depth mi
2018-05-06 20:22:12	2.6	0.2
2018-05-06 19:11:21	2.6	2.9
2018-05-06 18:11:14	2.5	-0.1
2018-05-06 18:07:46	2.7	-0.4
2018-05-06 17:59:58	2.8	2
2018-05-06 16:28:19	3	1.2
2018-05-06 15:49:02	2.8	5.2
2018-05-06 14:28:40	3	3.6

(USGS Volcano Hazards Program, May 2018)

Design Question: Earthquakes



393 Earthquakes Visible (of 393)

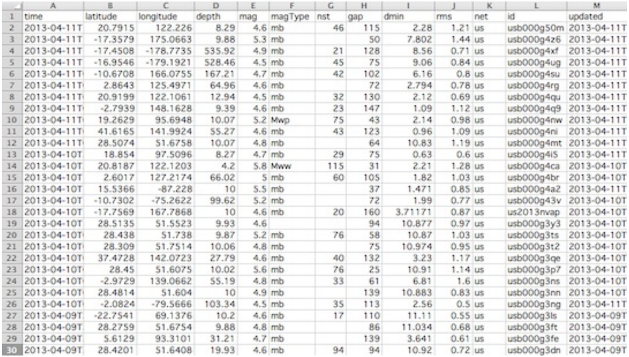
Zoom in to view lower magnitude quakes.

Date/Time local	Mag.	Depth mi
2018-05-06 20:22:12	2.6	0.2
2018-05-06 19:11:21	2.6	2.9
2018-05-06 18:11:14	2.5	-0.1
2018-05-06 18:07:46	2.7	-0.4
2018-05-06 17:59:58	2.8	2
2018-05-06 16:28:19	3	1.2
2018-05-06 15:49:02	2.8	5.2
2018-05-06 14:28:40	3	3.6
2018-05-06	3.3	0.4

(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.

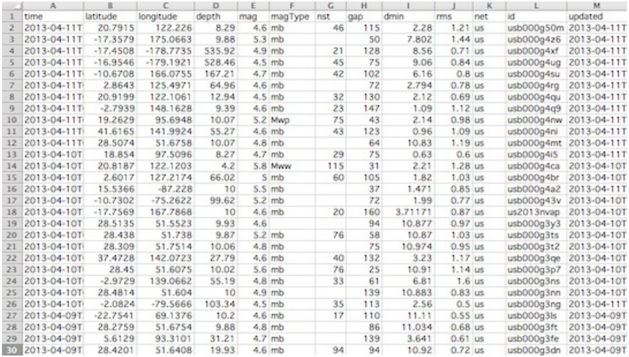
Design Question: Earthquakes



	A	B	C	D	E	F	G	H	I	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	us	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	2013-04-11T	-2.7939	148.1628	9.39	4.6	mb	23	147	1.09	1.12	us	usb000g4q9	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	2013-04-11T	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	Mww	115	31	2.21	1.28	us	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	10	5.5	mb		37	1.471	0.85	us	usb000g4a2	2013-04-11T
17	2013-04-10T	-10.7302	-75.2622	99.62	5.2	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	usb000g3qe	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	4.7	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)

Design Question: Earthquakes

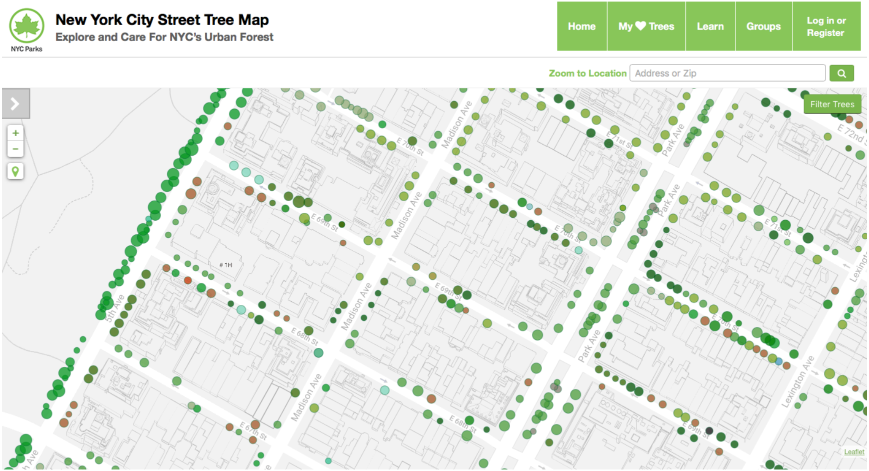


	A	B	C	D	E	F	G	H	I	J	K	L	M
1	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net	id	updated
2	2013-04-11T	20.7915	122.226	8.29	4.6	mb	46	115	2.28	1.21	us	usb000g50m	2013-04-11T
3	2013-04-11T	-17.3579	175.0663	9.88	5.3	mb		50	7.802	1.44	us	usb000g4z6	2013-04-11T
4	2013-04-11T	-17.4508	-178.7735	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	2013-04-11T	-2.7939	148.1628	9.39	4.6	mb	23	147	1.09	1.12	us	usb000g4q9	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	2013-04-11T	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	Mww	115	31	2.21	1.28	us	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	10	5.5	mb		37	1.471	0.85	us	usb000g4a2	2013-04-11T
17	2013-04-10T	-10.7302	-75.2622	99.62	5.2	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	usb000g3qe	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	4.7	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)

- Demo `turtles` solution.

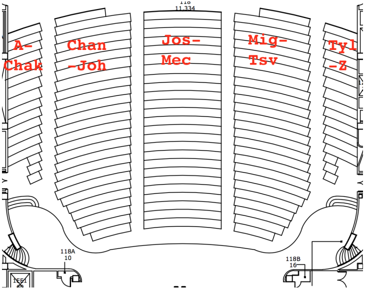
Design Question: Street Trees



New York City Street Tree Map

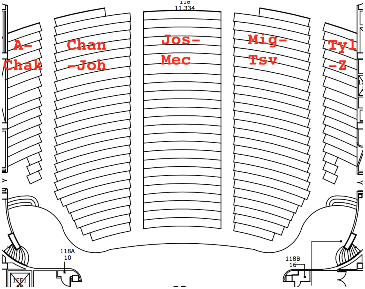
Explore and Care For NYC's Urban Forest

Mock Final



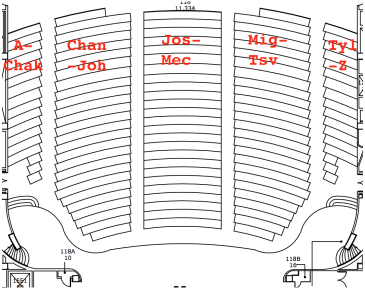
- Practice exam: the same format as the final (except 1, not full 2 hours).

Mock Final



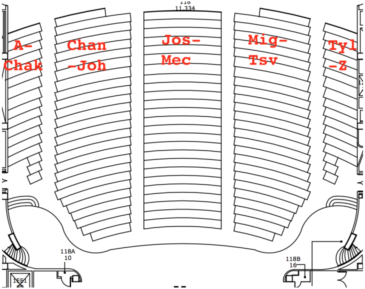
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).

Mock Final



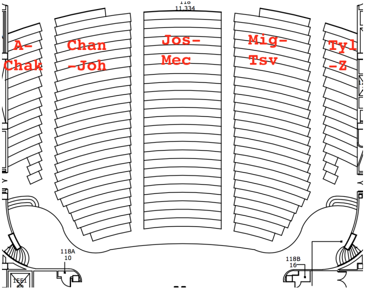
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.

Mock Final



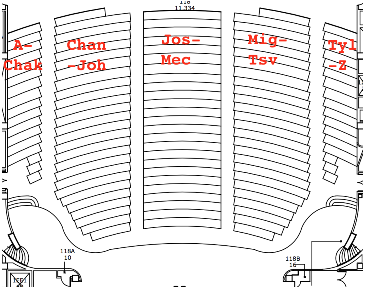
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.
  - Seating: every other seat, every other row.

Mock Final



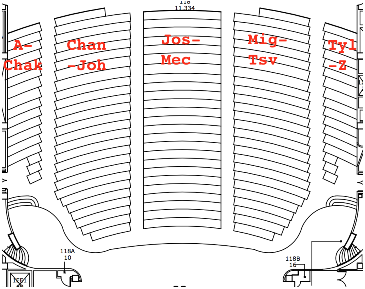
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.
  - Seating: every other seat, every other row.
  - Sign in as you enter (clipboards for each section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

Mock Final



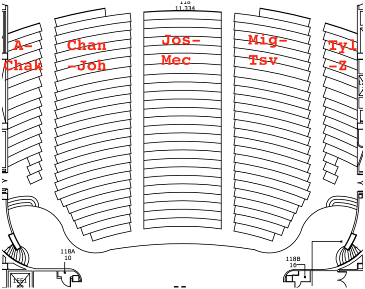
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.
  - Seating: every other seat, every other row.
  - Sign in as you enter (clipboards for each section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

Mock Final



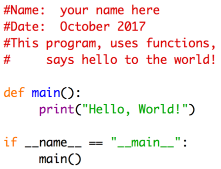
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.
    - Seating: every other seat, every other row.
    - Sign in as you enter (clipboards for each section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

Mock Final



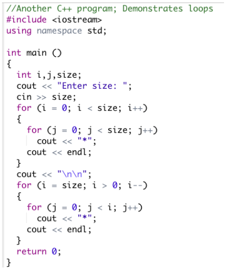
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating by last name: A-Chak, Chan-Joh, Jos-Mec, Mig-Tsv, Tyl-Z.
    - Seating: every other seat, every other row.
    - Sign in as you enter (clipboards for each section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

Final Review Questions



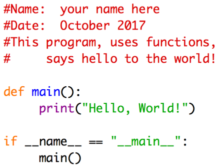
```
#Name:  your name here
#Date:  October 2017
#This program, uses functions,
#     says hello to the world!

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

if __name__ == "__main__":
     main()
```

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

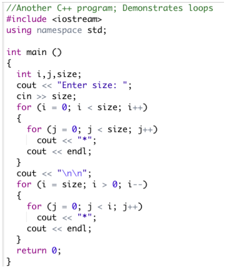


```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;

int main ()
{
  int i,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;
}
```

Final Review Questions



```
#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()
```

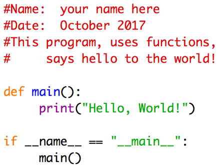


```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;

int main ()
{
  int i,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;
}
```

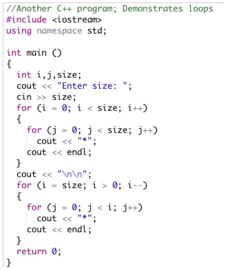
Plan: Alternate between working in pairs and sketching solutions (until time runs out):

- Definite Loops in Python & C++

Final Review Questions



```
#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()
```

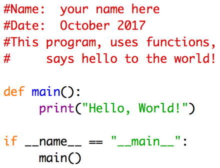


```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;

int main ()
{
  int i,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;
}
```

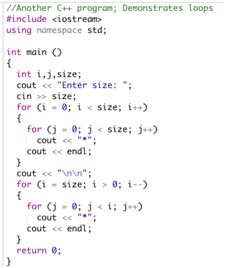
Plan: Alternate between working in pairs and sketching solutions (until time runs out):

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

Final Review Questions



```
#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,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;
}
```

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++

In Pairs or Triples: Definite 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)
  ```

- *C++: what is the output?*
  ```
  int i;
  for (i = 2017; i > 2000; i = i - 2)
      cout << "Year is " << i << endl)
  ```

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

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

In Pairs or Triples: Definite Loops in Python & C++

- *Python: what is the output?*

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

In Pairs or Triples: Definite Loops in Python & C++

- *C++: what is the output?*

```
int i;
for (i = 2017; i > 2000; i = i - 2)
    cout << "Year is " << i << endl)
```

In Pairs or Triples: Definite Loops in Python & C++

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

In Pairs or Triples: Definite Loops in Python & C++

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

In Pairs or Triples: Conditionals in Python & C++

- *Python: what is the output?*

```
year = 2016
if year % 4 == 0 and \
    (not (year % 100 == 0) or (year % 400 == 0)):
    print("Leap!!")
```

- *C++: what is the output?*

```
int i = 13;
if ((i % 2 == 1) && (i % 3 != 0) && !(i % 5 == 0))
    cout << i << " is not divisible!" << endl;
```

- *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).

In Pairs or Triples: Conditionals in Python & C++

- *Python: what is the output?*

```
year = 2016
if year % 4 == 0 and \
    (not (year % 100 == 0) or (year % 400 == 0)):
    print("Leap!!")
```

In Pairs or Triples: Conditionals in Python & C++

- *C++: what is the output?*

```
int i = 13;
if ((i % 2 == 1) && (i % 3 != 0) && !(i % 5 == 0))
     cout << i << " is not divisible!" << endl;
```

In Pairs or Triples: Conditionals in Python & C++

- *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.*

In Pairs or Triples: Conditionals in Python & C++

- *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:

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;
} 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.

In Pairs or Triples: Indefinite Loops in Python & C++

- *Python: what is the output?*

```
bal = 100
while bal < 200:
    print("Balance", bal)
    bal = bal + 0.1*bal
```

In Pairs or Triples: Indefinite Loops in Python & C++

- *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);
```

In Pairs or Triples: Indefinite Loops in Python & C++

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

In Pairs or Triples: Indefinite Loops in Python & C++

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