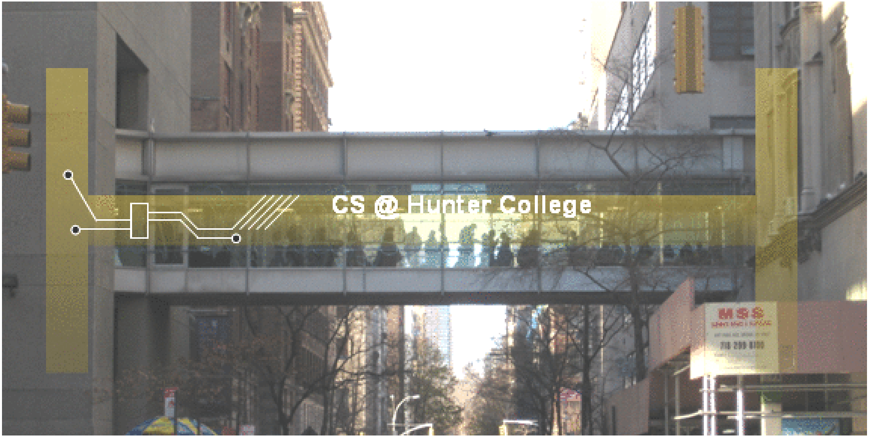
CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

Announcements

- Mock exam next week.
  Final exam: 13 days!

Announcements

- Mock exam next week.
  Final exam: 13 days!
- We end lecture with a survey of computing research and tech.

  Today: Adrienne Posner,
  Program Manager, Google

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Credit/NoCredit?

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Credit/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Credit/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Credit/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?
  First: it gives unfair advantage & is immoral.

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Credit/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?
  First: it gives unfair advantage & is immoral.
  Second: it's a standard question on faculty references.

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Credit/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?
  First: it gives unfair advantage & is immoral.
  Second: it's a standard question on faculty references.
  Industry & graduate schools hate it: don't want someone who falsifies work.

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Credit/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?
  First: it gives unfair advantage & is immoral.
  Second: it's a standard question on faculty references.
  Industry & graduate schools hate it: don't want someone who falsifies work.
- Why isn't Prof. St. John teaching next term?

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Credit/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?
  First: it gives unfair advantage & is immoral.
  Second: it's a standard question on faculty references.
  Industry & graduate schools hate it: don't want someone who falsifies work.
- Why isn't Prof. St. John teaching next term?
  She'll be on sabbatical for 2020, heading for New Zealand in February.

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Credit/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?
  First: it gives unfair advantage & is immoral.
  Second: it's a standard question on faculty references.
  Industry & graduate schools hate it: don't want someone who falsifies work.
- Why isn't Prof. St. John teaching next term?
  She'll be on sabbatical for 2020, heading for New Zealand in February.
  (Every 7 years, faculty can spend a year away, focused on research.)

Frequently Asked Questions

From lecture slips & email:

- I forgot/missed/want to change the final exam plans form. What do I do?
  No worries– fill out the pink slip (extras in lab) and give to the UTA's to upload.
- What's the best way to study for the final exam?
  The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
  Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Credit/NoCredit?
  It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Why do you care about cheating?
  First: it gives unfair advantage & is immoral.
  Second: it's a standard question on faculty references.
  Industry & graduate schools hate it: don't want someone who falsifies work.
- Why isn't Prof. St. John teaching next term?
  She'll be on sabbatical for 2020, heading for New Zealand in February.
  (Every 7 years, faculty can spend a year away, focused on research.) Kia Ora!

Frequently Asked Questions

From lecture slips & recitation sections.

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.

Frequently Asked Questions

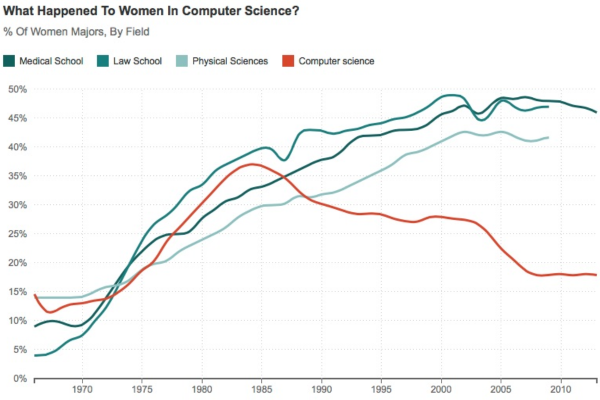
From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women. *Well, actually, there's not.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women. *Well, actually, there's not.*



What Happened To Women In Computer Science?

% Of Women Majors, By Field

Source: National Science Foundation, American Bar Association, American Association of Medical Colleges
Credit: Quoctrung Bui/NPR

(PlanetMoney)

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women. *Well, actually, there's not.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women. *Well, actually, there's not.*
- Yes, but I'm a guy. There's lots of guys.

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.
  Across CUNY there were < 1000 *students graduating with a tech degree (out of 24,000 annual graduates).*

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.
  Across CUNY there were < 1000 *students graduating with a tech degree (out of 24,000 annual graduates).*
  Versus Stanford where 90% take a CS course & half major in computer science.

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.
  Across CUNY there were < 1000 *students graduating with a tech degree (out of 24,000 annual graduates).*
  Versus Stanford where 90% take a CS course & half major in computer science.
- And how does this help me?

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.
  Across CUNY there were < 1000 *students graduating with a tech degree (out of 24,000 annual graduates).*
  Versus Stanford where 90% take a CS course & half major in computer science.
- And how does this help me?
  Redesigned CSci 127:

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.
  Across CUNY there were < 1000 *students graduating with a tech degree (out of 24,000 annual graduates).*
  Versus Stanford where 90% take a CS course & half major in computer science.
- And how does this help me?
  Redesigned CSci 127:
  - *Following Harvard: a rigorous & practical course to attract everyone.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.
  Across CUNY there were < 1000 *students graduating with a tech degree (out of 24,000 annual graduates).*
  Versus Stanford where 90% take a CS course & half major in computer science.
- And how does this help me?
  Redesigned CSci 127:
  - *Following Harvard: a rigorous & practical course to attract everyone.*
  - *Automated grading to give immediate feedback.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.
  Across CUNY there were < 1000 *students graduating with a tech degree (out of 24,000 annual graduates).*
  Versus Stanford where 90% take a CS course & half major in computer science.
- And how does this help me?
  Redesigned CSci 127:
  - *Following Harvard: a rigorous & practical course to attract everyone.*
  - *Automated grading to give immediate feedback.*
  - *Redirected resources to be student-focused (e.g. UTAs).*

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.
  Across CUNY there were < 1000 *students graduating with a tech degree (out of 24,000 annual graduates).*
  Versus Stanford where 90% take a CS course & half major in computer science.
- And how does this help me?
  Redesigned CSci 127:
  - *Following Harvard: a rigorous & practical course to attract everyone.*
  - *Automated grading to give immediate feedback.*
  - *Redirected resources to be student-focused (e.g. UTAs).*
  - *Dedicated lab space & new computers.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.
  Across CUNY there were < 1000 *students graduating with a tech degree (out of 24,000 annual graduates).*
  Versus Stanford where 90% take a CS course & half major in computer science.
- And how does this help me?
  Redesigned CSci 127:
  - *Following Harvard: a rigorous & practical course to attract everyone.*
  - *Automated grading to give immediate feedback.*
  - *Redirected resources to be student-focused (e.g. UTAs).*
  - *Dedicated lab space & new computers.*

  More students means more variety in upper division electives,

Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.
  Across CUNY there were < 1000 *students graduating with a tech degree (out of 24,000 annual graduates).*
  Versus Stanford where 90% take a CS course & half major in computer science.
- And how does this help me?
  Redesigned CSci 127:
  - *Following Harvard: a rigorous & practical course to attract everyone.*
  - *Automated grading to give immediate feedback.*
  - *Redirected resources to be student-focused (e.g. UTAs).*
  - *Dedicated lab space & new computers.*

  More students means more variety in upper division electives, more students with interests similar to yours,

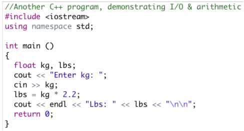
Frequently Asked Questions

From lecture slips & recitation sections.

- Why all the fuss about women in computer science? There's lots of women.
  Well, actually, there's not.
- Yes, but I'm a guy. There's lots of guys.
  Not compared to other schools.
  Across CUNY there were < 1000 *students graduating with a tech degree (out of 24,000 annual graduates).*
  Versus Stanford where 90% take a CS course & half major in computer science.
- And how does this help me?
  Redesigned CSci 127:
  - *Following Harvard: a rigorous & practical course to attract everyone.*
  - *Automated grading to give immediate feedback.*
  - *Redirected resources to be student-focused (e.g. UTAs).*
  - *Dedicated lab space & new computers.*

  More students means more variety in upper division electives, more students with interests similar to yours, and more links to research and industry.

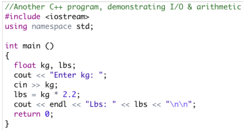
Today's Topics



```
//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;
}
```

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

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;
}
```

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

CS Surveys Talk: Google

Adrienne Posner

CS Surveys Talk: Google

Adrienne Posner

- What Adrienne does and loves about her job.

CS Surveys Talk: Google

Adrienne Posner

- What Adrienne does and loves about her job.
- Design challenge: work in pairs or triples on project management challenge.

Lecture Slip: Project Management

Jenn, Adrienne, & Nina

Lecture Slip: Project Management

Jenn, Adrienne, & Nina

- Design challenge: work in pairs or triples on project management challenge.

Lecture Slip: Project Management

Jenn, Adrienne, & Nina

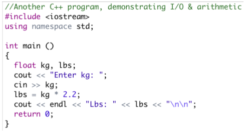
- Design challenge: work in pairs or triples on project management challenge.
- In computer science jargon, this is a "scheduling with constraints" problem.

Lecture Slip: Project Management

Jenn, Adrienne, & Nina

- Design challenge: work in pairs or triples on project management challenge.
- In computer science jargon, this is a "scheduling with constraints" problem.
- Similar questions in network flow, operating systems, and other resource-constrained settings.

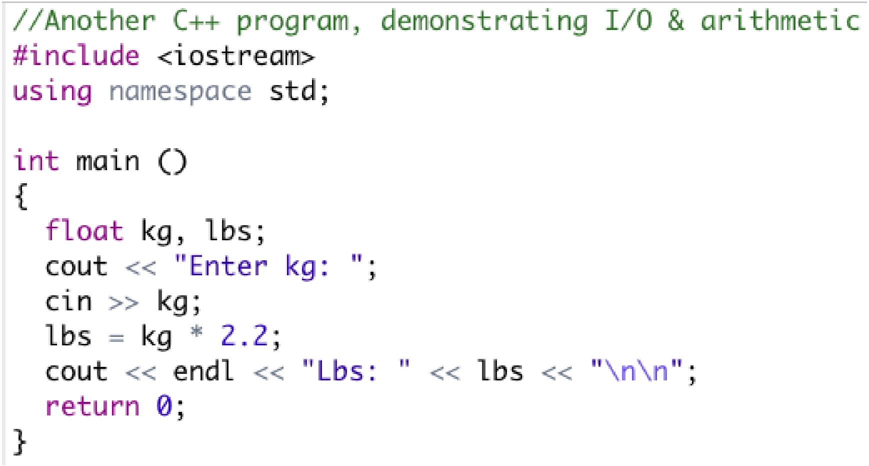
Today's Topics



```
//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;
}
```

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

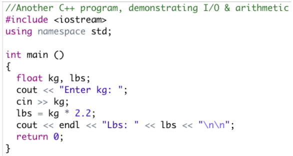
Recap: Basic Form & I/O in C++



```
//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;
}
```

Recap: Basic Form & I/O in C++

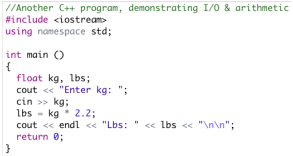
- Efficient for systems programming.



```
//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;
}
```

Recap: Basic Form & I/O in C++

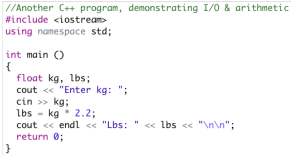
- Efficient for systems programming.
- Programs are organized in functions.



```
//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;
}
```

Recap: Basic Form & I/O in C++

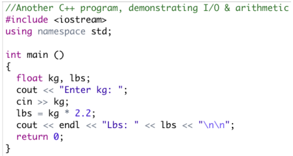
- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables:



```
//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;
}
```

Recap: Basic Form & I/O in C++

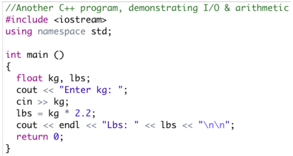
- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`



```
//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;
}
```

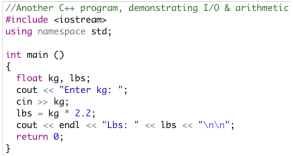
Recap: Basic Form & I/O in C++

- 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;
}
```

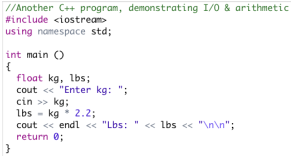
Recap: Basic Form & I/O in C++



```
//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, ...`

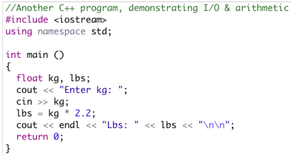
Recap: Basic Form & I/O in C++



```
//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:

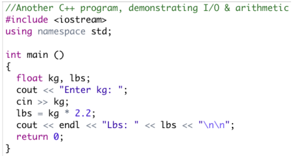
Recap: Basic Form & I/O in C++



```
//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!!";`

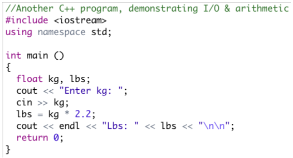
Recap: Basic Form & I/O in C++



```
//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:

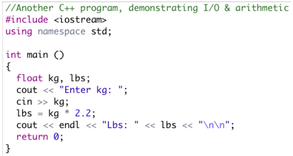
Recap: Basic Form & I/O in C++



```
//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;`

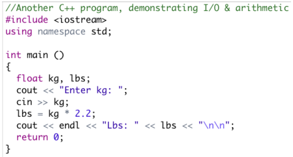
Recap: Basic Form & I/O in C++



```
//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:

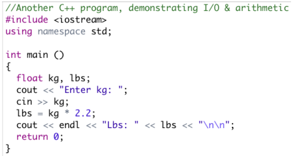
Recap: Basic Form & I/O in C++



```
//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;`

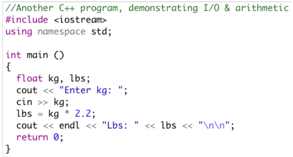
Recap: Basic Form & I/O in C++



```
//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:

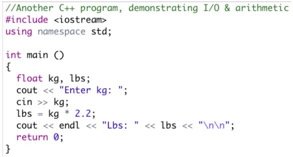
Recap: Basic Form & I/O in C++



```
//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++) {...}`

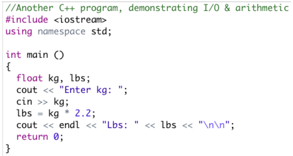
Recap: Basic Form & I/O in C++



```
//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 '}'.

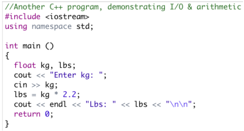
Recap: Basic Form & I/O in C++



```
//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 ';'.

Today's Topics

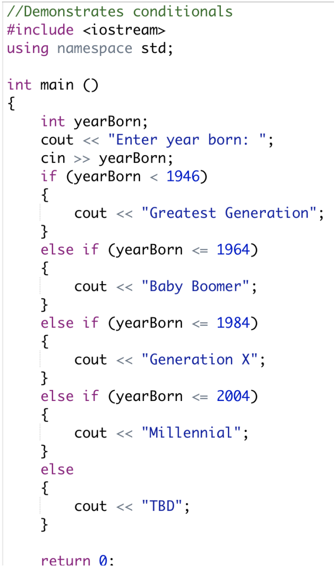


```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

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

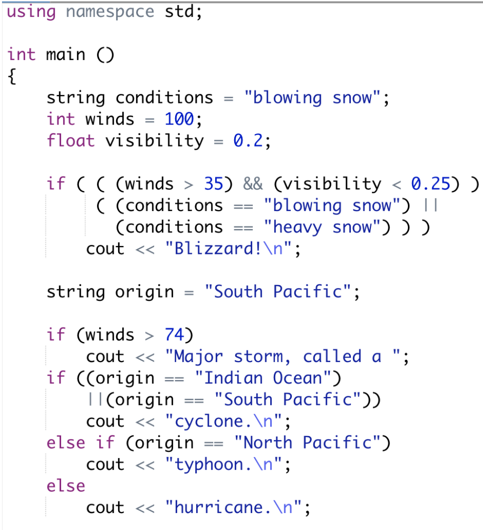
- CS Survey
- Recap: I/O & Definite Loops in C++
- **Conditionals in C++**
- Indefinite Loops in C++
- Recap: C++ & Python

In Pairs or Triples:

Predict what the following pieces of code will do:

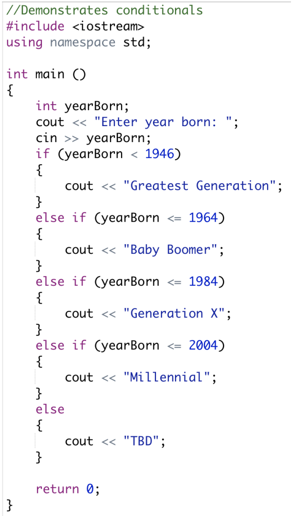


```
//Demonstrates conditionals
#include <iostream>
using namespace std;

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";
```

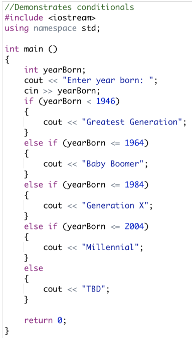
C++ Demo



```
//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 (`||`)

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

Logical Operators in C++

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

and (`&&`)

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

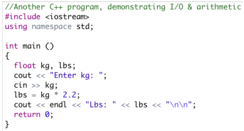
or (`||`)

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

not (`!`)

	in1	*returns:*
!	False	True
!	True	False

Today's Topics

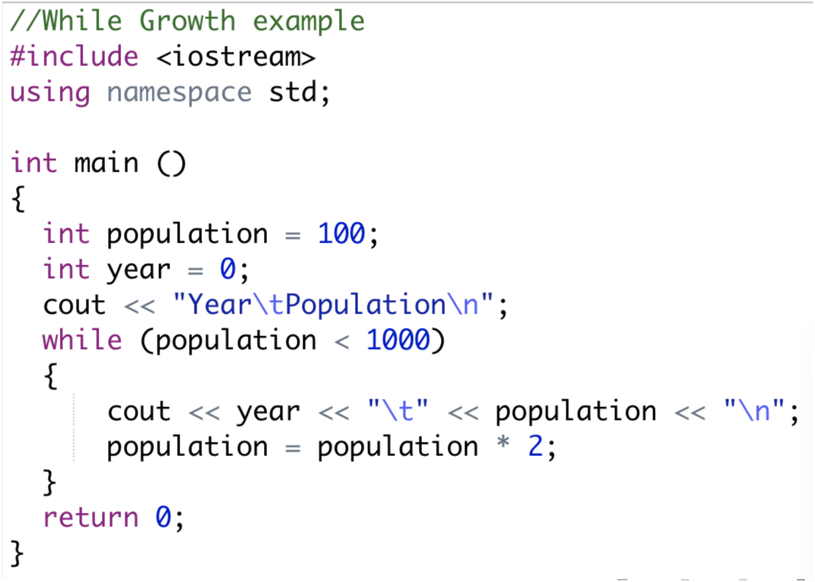


```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

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

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

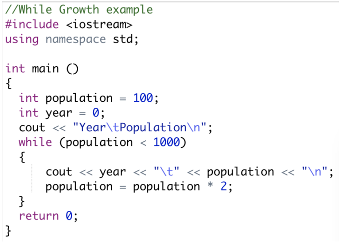
In Pairs or Triples:

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;
}
```

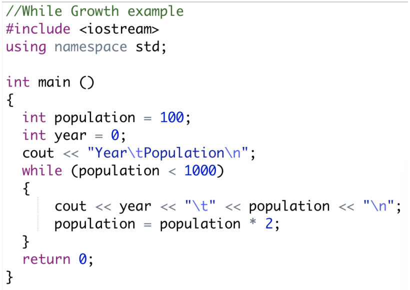
C++ Demo



```
//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:

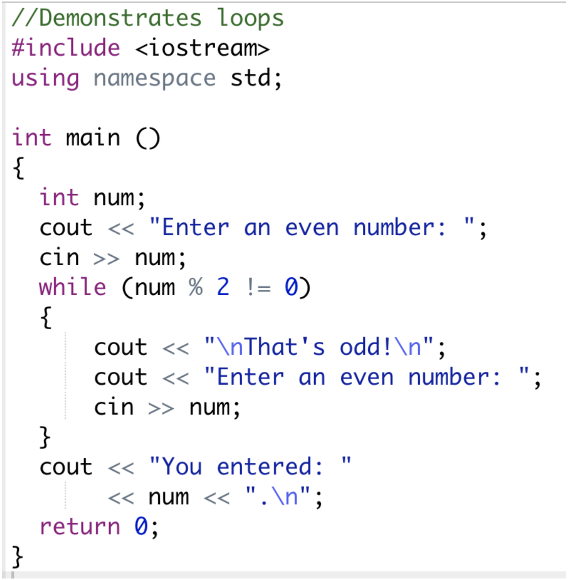
`while` (*logical expression*)
{

command1;
command2;
command3;
...

}

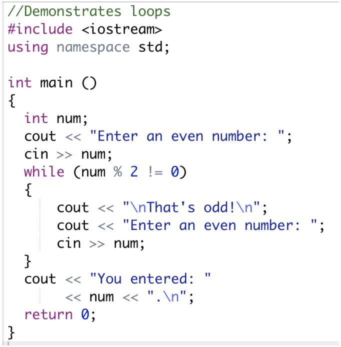
In Pairs or Triples:

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;
}
```

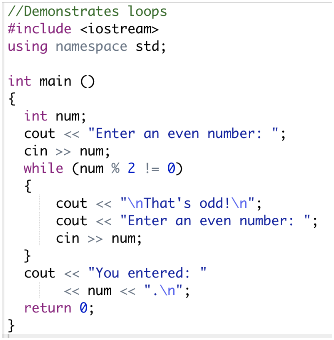
C++ Demo



```
//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:

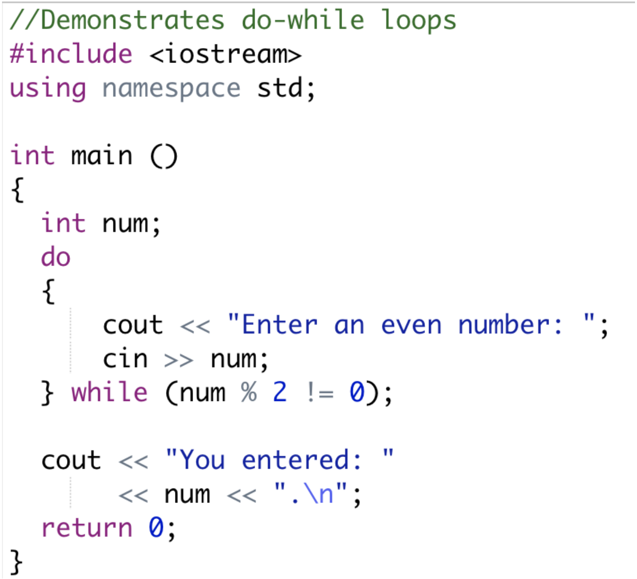
`while` (*logical expression*)
{

command1;
command2;
command3;
...

}

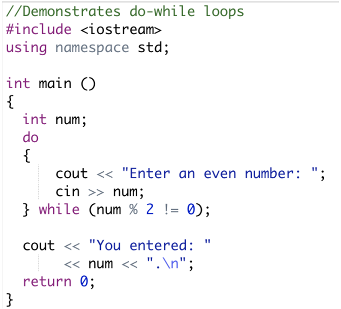
In Pairs or Triples:

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;
}
```

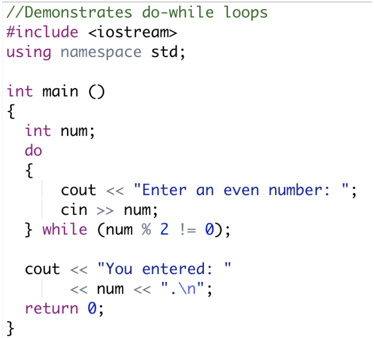
C++ Demo



```
//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`

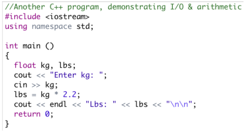


```cpp
//Demonstrates do-while loops
#include <iostream>
using namespace std;

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 );
```

Today's Topics

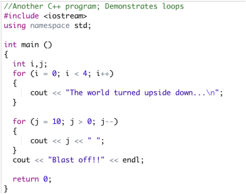


```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

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

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

Recap: C++ Control Structures

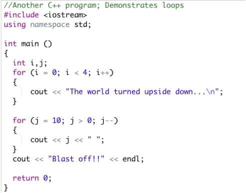
- I/O:



```
//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

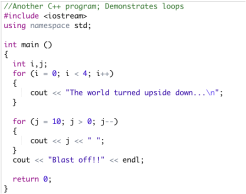
- I/O: `cin >> ...;`



```cpp
//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

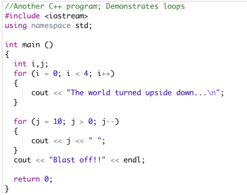
- I/O: `cin >> ...;` & `cout << ...;`



```
//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:

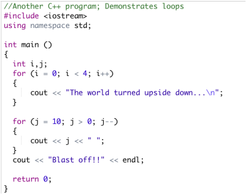


```cpp
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;

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

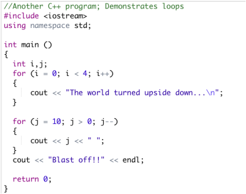
- I/O: `cin >> ...;` & `cout << ...;`
- Definite loops:

```
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



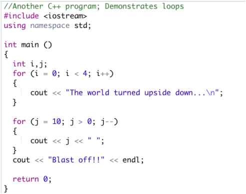
```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;

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:

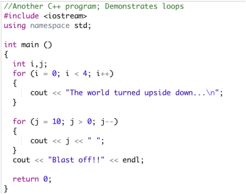
  ```
  for (i = 0; i < 10; i++)
  {
        ...
  }
  ```

- Conditionals:

  `if` (*logical expression*)

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

Recap: C++ Control Structures

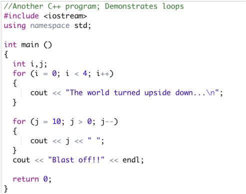


```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;

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)
  {
        ...
  }
  ```

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

- *Rewrite this program in C++:*

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

- *Rewrite this program in Python:*

```
#include <iostream>
using namespace std;
int main()
{
  for (int i = 1; i < 50; i++)
  {
    cout << i << endl;
  }
  return 0;
}
```

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

- *Rewrite this program in C++:*

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

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

- *Rewrite this program in C++:*

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

```
#include <iostream>
using namespace std;
```

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

- *Rewrite this program in C++:*

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

```
#include <iostream>
using namespace std;
int main()
```

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

- *Rewrite this program in C++:*

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

```
#include <iostream>
using namespace std;
int main()
{
```

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

- *Rewrite this program in C++:*

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

```
#include <iostream>
using namespace std;
int main()
{
  for (int i = 2017; i >= 2000; i=i-2)
```

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

- *Rewrite this program in C++:*

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

```
#include <iostream>
using namespace std;
int main()
{
  for (int i = 2017; i >= 2000; i=i-2)
  {
    cout << "Year is" << i << endl;
```

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

- *Rewrite this program in C++:*

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

```
#include <iostream>
using namespace std;
int main()
{
  for (int i = 2017; i >= 2000; i=i-2)
  {
    cout << "Year is" << i << endl;
  }
  return 0;
}
```

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

- *Rewrite this program in Python:*

```
#include <iostream>
using namespace std;
int main()
{
  for (int i = 1; i < 50; i++)
  {
    cout << i << endl;
  }
  return 0;
}
```

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

- *Rewrite this program in Python:*

```
#include <iostream>
using namespace std;
int main()
{
  for (int i = 1; i < 50; i++)
  {
    cout << i << endl;
  }
  return 0;
}
```

```
for i in range(1, 50):
```

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

- *Rewrite this program in Python:*

```
#include <iostream>
using namespace std;
int main()
{
  for (int i = 1; i < 50; i++)
  {
    cout << i << endl;
  }
  return 0;
}
```

```
for i in range(1, 50):
    print(i)
```

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

- *Write a C++ program that asks the user the number of times they plan to ride transit this week. Your program should then print if it is cheaper to buy single ride metro cards or 7-day unlimited card.*
  (The 7-day card is $33.00, and the cost of single ride, with bonus, is $2.75).

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

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!!")
print("Year")  year = 2016
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Prints: `Leap!`
`Year`

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

- *Your program should then print if it is cheaper to buy single ride metro cards ($2.75 per ride) or 7-day unlimited card ($33.00).*

  ```
  #include <iostream>
  using namespace std;
  ```

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

- *Your program should then print if it is cheaper to buy single ride metro cards ($2.75 per ride) or 7-day unlimited card ($33.00).*

```
#include <iostream>
using namespace std;
int main()
```

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

- *Your program should then print if it is cheaper to buy single ride metro cards ($2.75 per ride) or 7-day unlimited card ($33.00).*

```
#include <iostream>
using namespace std;
int main()
{
  int rides;
```

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

- *Your program should then print if it is cheaper to buy single ride metro cards ($2.75 per ride) or 7-day unlimited card ($33.00).*

```
#include <iostream>
using namespace std;
int main()
{
  int rides;
  cout << "Enter number of rides:";
```

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

- *Your program should then print if it is cheaper to buy single ride metro cards ($2.75 per ride) or 7-day unlimited card ($33.00).*

```
#include <iostream>
using namespace std;
int main()
{
  int rides;
  cout << "Enter number of rides:";
  cin >> rides;
```

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

- *Your program should then print if it is cheaper to buy single ride metro cards ($2.75 per ride) or 7-day unlimited card ($33.00).*

```
#include <iostream>
using namespace std;
int main()
{
  int rides;
  cout << "Enter number of rides:";
  cin >> rides;
  if (2.75 * rides < 33.00)
```

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

- *Your program should then print if it is cheaper to buy single ride metro cards ($2.75 per ride) or 7-day unlimited card ($33.00).*

```
#include <iostream>
using namespace std;
int main()
{
  int rides;
  cout << "Enter number of rides:";
  cin >> rides;
  if (2.75 * rides < 33.00)
  {
    cout << "Cheaper to buy single ride metro cards.\n";
  }
```

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

- *Your program should then print if it is cheaper to buy single ride metro cards ($2.75 per ride) or 7-day unlimited card ($33.00).*

```
#include <iostream>
using namespace std;
int main()
{
  int rides;
  cout << "Enter number of rides:";
  cin >> rides;
  if (2.75 * rides < 33.00)
  {
    cout << "Cheaper to buy single ride metro cards.\n";
  }
  else
```

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

- *Your program should then print if it is cheaper to buy single ride metro cards ($2.75 per ride) or 7-day unlimited card ($33.00).*

```
#include <iostream>
using namespace std;
int main()
{
  int rides;
  cout << "Enter number of rides:";
  cin >> rides;
  if (2.75 * rides < 33.00)
  {
    cout << "Cheaper to buy single ride metro cards.\n";
  }
  else
  {
    cout << "Cheaper to buy 7-day unlimited card.\n";
  }
```

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

- *Your program should then print if it is cheaper to buy single ride metro cards ($2.75 per ride) or 7-day unlimited card ($33.00).*

```
#include <iostream>
using namespace std;
int main()
{
  int rides;
  cout << "Enter number of rides:";
  cin >> rides;
  if (2.75 * rides < 33.00)
  {
    cout << "Cheaper to buy single ride metro cards.\n";
  }
  else
  {
    cout << "Cheaper to buy 7-day unlimited card.\n";
  }
  return 0;
}
```

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

- 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++

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

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

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

```
s = ""
```

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

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

```
s = ""
while s == "":
```

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

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

```
s = ""
while s == "":
  s = input("Enter a non-empty string:  ")
```

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

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

```
s = ""
while s == "":
  s = input("Enter a non-empty string:  ")
print("You entered:  ", s)
```

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

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

  ```
  s = ""
  while s == "":
    s = input("Enter a non-empty string:  ")
  print("You entered:  ", s)
  ```

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

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

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

```
s = ""
while s == "":
  s = input("Enter a non-empty string:  ")
print("You entered:  ", s)
```

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

```
#include <iostream>
using namespace std;
```

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

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

```python
s = ""
while s == "":
  s = input("Enter a non-empty string:  ")
print("You entered:  ", s)
```

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

```cpp
#include <iostream>
using namespace std;
int main()
```

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

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

```
s = ""
while s == "":
  s = input("Enter a non-empty string:  ")
print("You entered:  ", s)
```

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

```
#include <iostream>
using namespace std;
int main()
{
  int num = 0;
```

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

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

```
s = ""
while s == "":
  s = input("Enter a non-empty string:  ")
print("You entered:  ", s)
```

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

```
#include <iostream>
using namespace std;
int main()
{
  int num = 0;
  while (num % 2 == 0)
```

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

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

```
s = ""
while s == "":
  s = input("Enter a non-empty string:  ")
print("You entered:  ", s)
```

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

```
#include <iostream>
using namespace std;
int main()
{
  int num = 0;
  while (num % 2 == 0)
  {
    cout << "Enter an odd number:";
```

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

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

```
s = ""
while s == "":
  s = input("Enter a non-empty string:  ")
print("You entered:  ", s)
```

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

```
#include <iostream>
using namespace std;
int main()
{
  int num = 0;
  while (num % 2 == 0)
  {
    cout << "Enter an odd number:";
    cin >> num;
```

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

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

```
s = ""
while s == "":
  s = input("Enter a non-empty string:  ")
print("You entered:  ", s)
```

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

```
#include <iostream>
using namespace std;
int main()
{
  int num = 0;
  while (num % 2 == 0)
  {
    cout << "Enter an odd number:";
    cin >> num;
  }
```

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

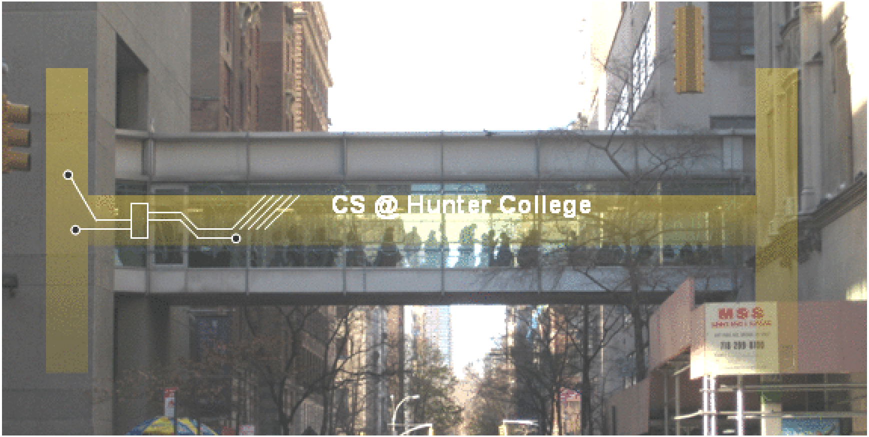
- Write Python code that repeatedly prompts for a non-empty string.

```
s = ""
while s == "":
  s = input("Enter a non-empty string:  ")
print("You entered:  ", s)
```

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

```
#include <iostream>
using namespace std;
int main()
{
  int num = 0;
  while (num % 2 == 0)
  {
    cout << "Enter an odd number:";
    cin >> num;
  }
  return 0;
}
```

Lecture Slips & Writing Boards



- Hand lecture slips to UTA's & return writing boards as you leave...