

# CSci 127: Introduction to Computer Science



[hunter.cuny.edu/csci](http://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.*

# Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?

*We're part of a pilot program for prospective students.*

- I have a conflict with the final– what should I do?

# Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?

*We're part of a pilot program for prospective students.*

- I have a conflict with the final– what should I do?

*Fill in the pink slip with when you will take the final.*

# Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?

*We're part of a pilot program for prospective students.*

- I have a conflict with the final– what should I do?

*Fill in the pink slip with when you will take the final.*

*Note: if you are not taking it at the assigned time, include reason for dean's approval for the non-standard time.*

# Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?

*We're part of a pilot program for prospective students.*

- I have a conflict with the final– what should I do?

*Fill in the pink slip with when you will take the final.*

*Note: if you are not taking it at the assigned time, include reason for dean's approval for the non-standard time.*

- 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.*

- I have a conflict with the final– what should I do?

*Fill in the pink slip with when you will take the final.*

*Note: if you are not taking it at the assigned time, include reason for dean's approval for the non-standard time.*

- 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 & recitation sections.

- Who/why all the visitors?

*We're part of a pilot program for prospective students.*

- I have a conflict with the final– what should I do?

*Fill in the pink slip with when you will take the final.*

*Note: if you are not taking it at the assigned time, include reason for dean's approval for the non-standard time.*

- 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.*

- 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.*

- I have a conflict with the final– what should I do?

*Fill in the pink slip with when you will take the final.*

*Note: if you are not taking it at the assigned time, include reason for dean's approval for the non-standard time.*

- 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.*

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

► *Majors: CSci 135/136 (C++, MWTh 12:10-1pm + section) & CSci 150 (Discrete Structures, MTh 1:10-2:25pm + section)*

# Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?

*We're part of a pilot program for prospective students.*

- I have a conflict with the final– what should I do?

*Fill in the pink slip with when you will take the final.*

*Note: if you are not taking it at the assigned time, include reason for dean's approval for the non-standard time.*

- 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.*

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

- ▶ Majors: *CSci 135/136 (C++, MWTh 12:10-1pm + section) & CSci 150 (Discrete Structures, MTh 1:10-2:25pm + section)*
- ▶ Minors: *CSci 133 (More Python: multiple times) & CSci 232 (Databases, multiple times)*

# Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?

*We're part of a pilot program for prospective students.*

- I have a conflict with the final– what should I do?

*Fill in the pink slip with when you will take the final.*

*Note: if you are not taking it at the assigned time, include reason for dean's approval for the non-standard time.*

- 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.*

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

- ▶ Majors: *CSci 135/136 (C++, MWTh 12:10-1pm + section) & CSci 150 (Discrete Structures, MTh 1:10-2:25pm + section)*
- ▶ Minors: *CSci 133 (More Python: multiple times) & CSci 232 (Databases, multiple times)*

- What's a mock exam? I see it on the webpage...

# Frequently Asked Questions

From lecture slips & recitation sections.

- Who/why all the visitors?

*We're part of a pilot program for prospective students.*

- I have a conflict with the final– what should I do?

*Fill in the pink slip with when you will take the final.*

*Note: if you are not taking it at the assigned time, include reason for dean's approval for the non-standard time.*

- 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.*

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

- ▶ Majors: *CSci 135/136 (C++, MWTh 12:10-1pm + section) & CSci 150 (Discrete Structures, MTh 1:10-2:25pm + section)*
- ▶ Minors: *CSci 133 (More Python: multiple times) & CSci 232 (Databases, multiple times)*

- What's a mock exam? I see it on the webpage...

*It's a practice exam that we're holding on 14 May.*

*More details at the end of lecture.*

# Announcements



- Three handouts today:
  - ▶ Lecture slip,
  - ▶ Final exam plans, and
  - ▶ Exam from last term.

# Today's Topics



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

# Today's Topics



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

## In Pairs or Triples:

- Using what you know from Python, predict what the C++ code will do:

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# onlinegdb demo

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

(Demo with onlinegdb)

# Introduction to C++

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

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# Introduction to C++

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

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

# Introduction to C++

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

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

# Introduction to C++

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

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

# Introduction to C++

- Programs are organized in functions.

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# Introduction to C++

- Programs are organized in functions.

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

Example:

# Introduction to C++

- Programs are organized in functions.

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

Example:

```
int main()
```

# Introduction to C++

- Programs are organized in functions.

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

Example:

```
int main()
{
```

# Introduction to C++

- Programs are organized in functions.

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

Example:

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

# Introduction to C++

- Programs are organized in functions.

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# Introduction to C++

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

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# Introduction to C++

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

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# Introduction to C++

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

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# Introduction to C++

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

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# Introduction to C++

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

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# Introduction to C++

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

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# Introduction to C++

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

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# Introduction to C++

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

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11    return 0;
12 }
```

# Introduction to C++

- Programs are organized in functions.

- Variables must be **declared**:

```
int num;
```

- Many types available:

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

- Semicolons separate commands:

```
num = 5; more = 2*num;
```

- To print, we'll use cout <<:

```
cout << "Hello!!";
```

- To get input, we'll use cin >>:

```
cin >> num;
```

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11
12 }
```

# Introduction to C++

- Programs are organized in functions.

- Variables must be **declared**:

```
int num;
```

- Many types available:

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

- Semicolons separate commands:

```
num = 5; more = 2*num;
```

- To print, we'll use cout <<:

```
cout << "Hello!!";
```

- To get input, we'll use cin >>:

```
cin >> num;
```

- To use those I/O functions, we put at the top of the program:

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11
12 }
```

# Introduction to C++

- Programs are organized in functions.

- Variables must be **declared**:

```
int num;
```

- Many types available:

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

- Semicolons separate commands:

```
num = 5; more = 2*num;
```

- To print, we'll use cout <<:

```
cout << "Hello!!";
```

- To get input, we'll use cin >>:

```
cin >> num;
```

- To use those I/O functions, we put at the top of the program:

```
#include <iostream>
```

```
using namespace std;
```

```
1 //Another C++ program, demonstrating variables
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     int year;
8     cout << "Enter a number: ";
9     cin >> year;
10    cout << "Hello " << year << "!!\n\n";
11
12 }
```

## In Pairs or Triples:

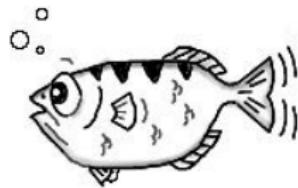
Predict what the following pieces of code will do:

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

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

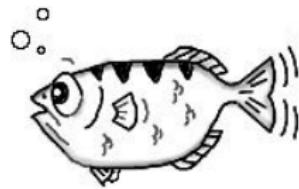
## Side Note: gdb

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



[gdb.org](http://gdb.org)

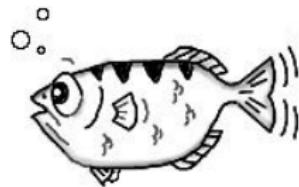
## Side Note: gdb



[gdb.org](http://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.

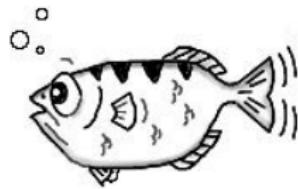
## Side Note: gdb



[gdb.org](http://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](http://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](http://onlinegdb.com)).

# C++ Demo

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

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

(Demo with onlinedbg)

## In Pairs or Triples...

Convert the C++ code to a **Python** program:

```
//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;
}
```

# Python Tutor

Convert the C++ code to a **Python program**:

```
//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;
}
```

(Write from scratch in `pythonTutor`.)

# Today's Topics



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

## In Pairs or Triples:

Predict what the following pieces of code will do:

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

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

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

    return 0;
}
```

# C++ Demo

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

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

(Demo with onlinegdb)

# Definite loops

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

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

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

    return 0;
}
```

General format:

```
for ( initialization ; test ; updateAction )
{
    command1;
    command2;
    command3;
    ...
}
```

# In Pairs or Triples:

Predict what the following pieces of code will do:

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

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

# C++ Demo

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

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

(Demo with onlinedbg)

# In Pairs or Triples:

Predict what the following pieces of code will do:

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

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

# C++ Demo

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

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

(Demo with onlinegdb)

# Lecture Slips

Which UTA have you spoken with most? Why?



Anna Lis



Antonio Bountouvas



Brian Campbell



Bryan Belmont



Camryn Buonamassa



David Yuen



Erin Williams



Ferdi Lesporis



Harry Wu



Mandy Yu



Michael Nurilov



Nick Szewczak



Nicky Cen



Owen Kunhardt



Parakram Basnet



Ralph Venté



Rhia Singh



Savannah Nester



Shaina Lowenthal



Shantel Dixon



Stephanie Yung



Stephen Milani



Such Singh



Thomas Joy



Tommi Ann Tsuruga



Vincent Zheng

# Lecture Slips

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

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

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

# Lecture Slips

Translate line-by-line:

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

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

# Recap: C++

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



# Recap: C++

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



# Recap: C++

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

- ▶ `cin >>`
- ▶ `cout <<`



# Recap: C++



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

- ▶ `cin >>`
- ▶ `cout <<`

- Definite loops:

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

# Recap: C++



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

```
for (i = 0; i < 10; i++) {  
    ...  
}
```
- Pass your lecture slip to the aisles for UTA's to collect.

# Today's Topics



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

# Final Exam: When



- The final exam is **Tuesday, 21 May, 9am-11am**, Assembly Hall (118 HN).

# Final Exam: When



- The final exam is **Tuesday, 21 May, 9am-11am**, Assembly Hall (118 HN).
- If you have a conflict, the alternative time is: Thursday, 16 May, 8:30-10:30am, 1001E HN.

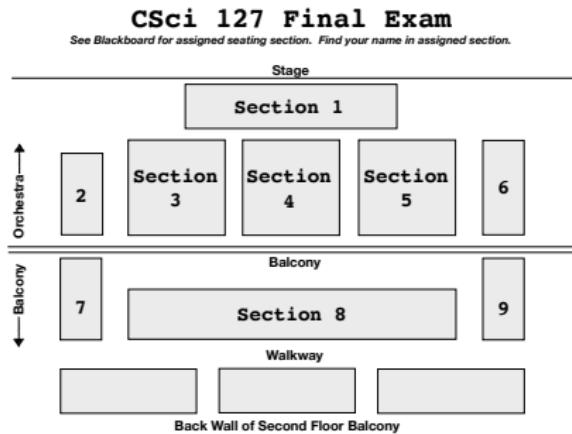
# Final Exam: When



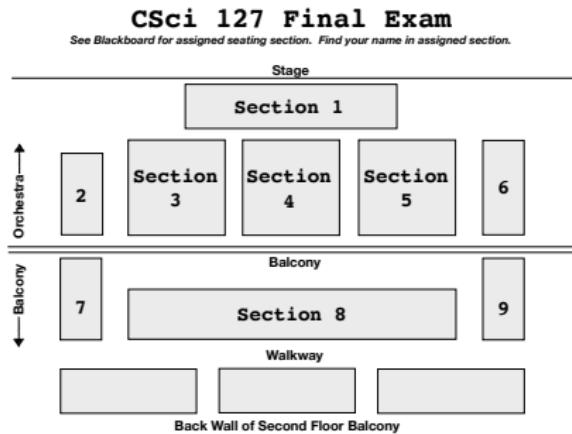
- The final exam is **Tuesday, 21 May, 9am-11am**, Assembly Hall (118 HN).
- If you have a conflict, the alternative time is: Thursday, 16 May, 8:30-10:30am, 1001E HN.
- If you have accommodations via the Accessibility Office, we will send the exam to their testing center.  
(Must complete by noon, Tuesday, 21 May.)

# Final Exam: Logistics

- Bring ID, note sheet, pencils or pens.

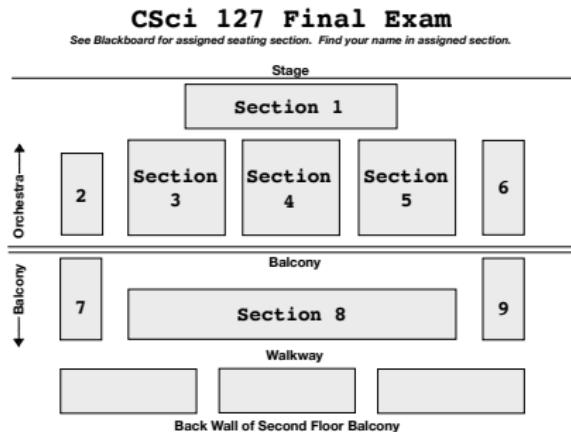


# Final Exam: Logistics



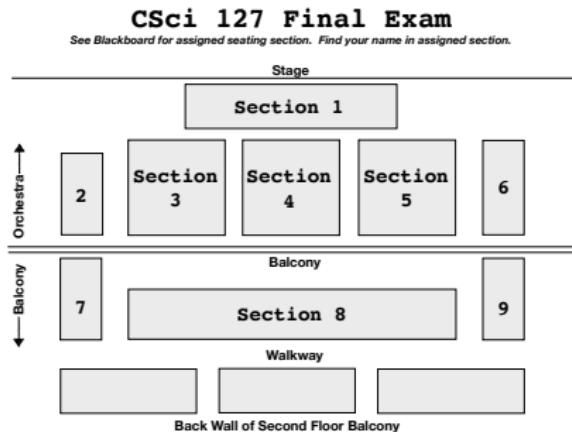
- Bring ID, note sheet, pencils or pens.
- Seating is assigned. See Blackboard for assignments.

# Final Exam: Logistics



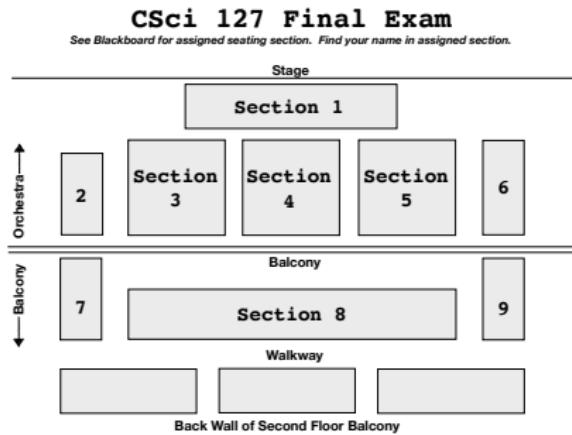
- Bring ID, note sheet, pencils or pens.
- Seating is assigned. See Blackboard for assignments.
- Sign out when you turn in your exam.

# Final Exam: Logistics



- Bring ID, note sheet, pencils or pens.
- Seating is assigned. See Blackboard for assignments.
- Sign out when you turn in your exam.
- Cannot leave during the first 45 minutes of the exam.

# Final Exam: Logistics



- Bring ID, note sheet, pencils or pens.
- Seating is assigned. See Blackboard for assignments.
- Sign out when you turn in your exam.
- Cannot leave during the first 45 minutes of the exam.
- Cannot start the exam after students start leaving.

# Final Exam: Format

- The exam is 2 hours long.

# Final Exam: Format

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

# Final Exam: Format

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

# Final Exam: Format

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

# Final Exam: Format

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

# Final Exam: Format

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

# Final Exam: Format

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

# Final Exam: Format

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

# Final Exam: Format

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

# Final Exam: Format

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

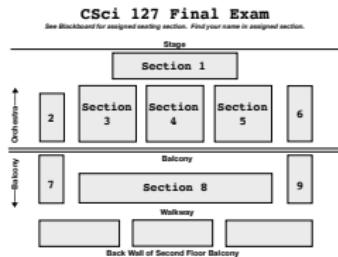
# Final Exam: Format

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

# Final Exam: Format

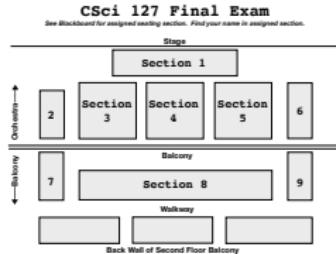
- The exam is 2 hours long.
- There are 4 different versions to discourage copying.
- It is on paper. No use of computers, phones, etc. allowed.
- You may have 1 piece of **8.5" x 11"** piece of paper.
  - ▶ With notes, examples, programs: what will help you on the exam.
  - ▶ No origami— it's distracting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

# Mock Final



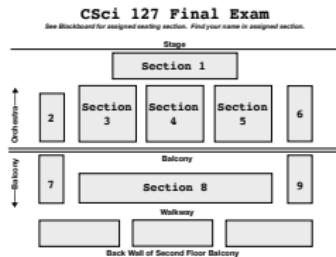
- Given in lecture on 14 May.

# Mock Final



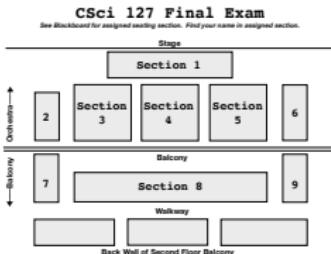
- Given in lecture on 14 May.
- Practice exam: the same format as the final (except 1, not full 2 hours).

# Mock Final



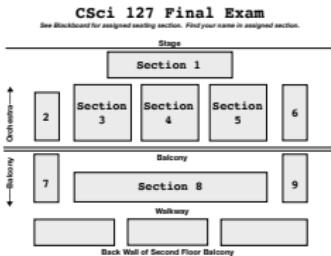
- Given in lecture on 14 May.
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).

# Mock Final



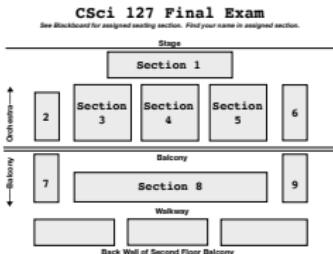
- Given in lecture on 14 May.
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating is assigned.

# Mock Final



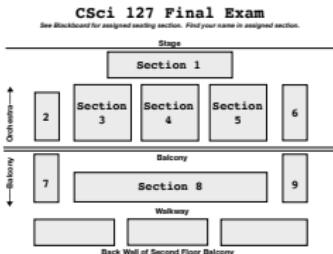
- Given in lecture on 14 May.
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating is assigned.
  - Assigned seating (see Blackboard): every other seat, every other row.

# Mock Final



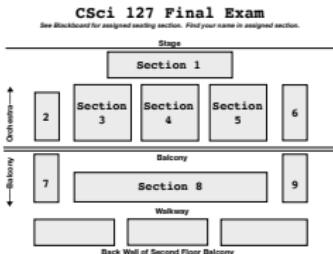
- Given in lecture on 14 May.
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating is assigned.
  - Assigned seating (see Blackboard): every other seat, every other row.
  - Fill out exam slip (pre-printed at your designated seat).

# Mock Final



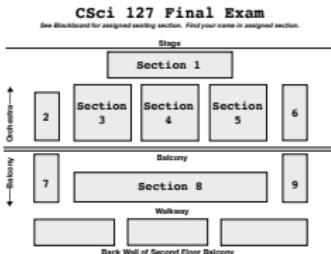
- Given in lecture on 14 May.
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating is assigned.
  - Assigned seating (see Blackboard): every other seat, every other row.
  - Fill out exam slip (pre-printed at your designated seat).
  - Sign out as you leave (clipboards for each section).

# Mock Final



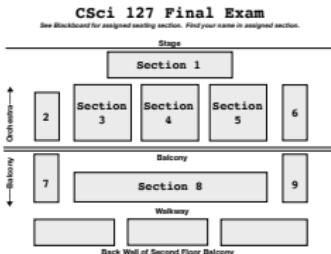
- Given in lecture on 14 May.
- Practice exam: the same format as the final (except 1, not full 2 hours).
- Bring ID & 1 page of notes (will check IDs during exam).
- Seating is assigned.
  - Assigned seating (see Blackboard): every other seat, every other row.
  - Fill out exam slip (pre-printed at your designated seat).
  - Sign out as you leave (clipboards for each section).
  - Cannot leave in first 30 minutes.

# Mock Final



- Given in lecture on 14 May.
- 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 is assigned.
  - ▶ Assigned seating (see Blackboard): every other seat, every other row.
  - ▶ Fill out exam slip (pre-printed at your designated seat).
  - ▶ Sign out as you leave (clipboards for each section).
  - ▶ Cannot leave in first 30 minutes.
- Lecture slip for that week: signing out at end of mock exam.

# Mock Final



- Given in lecture on 14 May.
- 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 is assigned.
  - ▶ Assigned seating (see Blackboard): every other seat, every other row.
  - ▶ Fill out exam slip (pre-printed at your designated seat).
  - ▶ Sign out as you leave (clipboards for each section).
  - ▶ Cannot leave in first 30 minutes.
- Lecture slip for that week: signing out at end of mock exam.
- Answer key will be available on webpage after lecture.

# Practice Quiz & Final Questions



- Lightning rounds:

# Practice Quiz & Final Questions



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

# Practice Quiz & Final Questions



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

# Practice Quiz & Final Questions



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

# Practice Quiz & Final Questions



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

# Practice Quiz & Final Questions



- Lightning rounds:
  - ▶ write as much you can for 60 seconds;
  - ▶ followed by answer; and
  - ▶ repeat.
- Past exams are on the webpage ([under Final Exam Information](#)).
- We'll start with Fall 18, Version 3 (hand-out).

# Writing Boards



- Return writing boards as you leave...