CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

• Please take a moment to fill out the **Teacher Evaluations**

- Please take a moment to fill out the Teacher Evaluations
- Your chance to provide feedback!

- Please take a moment to fill out the Teacher Evaluations
- Your chance to provide feedback!
- Responses are completely anonymous.

- Please take a moment to fill out the Teacher Evaluations
- Your chance to provide feedback!
- Responses are completely anonymous.
- **INCENTIVE:** if we reach 75% of submitted evaluations, I will drop the 6th lowest program grade.
- Smartphone: www.hunter.cuny.edu/mobilete

- Please take a moment to fill out the Teacher Evaluations
- Your chance to provide feedback!
- Responses are completely anonymous.
- **INCENTIVE:** if we reach 75% of submitted evaluations, I will drop the 6th lowest program grade.
- Smartphone: www.hunter.cuny.edu/mobilete
- Computer: www.hunter.cuny.edu/te

- Please take a moment to fill out the Teacher Evaluations
- Your chance to provide feedback!
- Responses are completely anonymous.
- **INCENTIVE:** if we reach 75% of submitted evaluations, I will drop the 6th lowest program grade.
- Smartphone: www.hunter.cuny.edu/mobilete
- Computer: www.hunter.cuny.edu/te

• Final Exam Monday 23 May

- Final Exam Monday 23 May
- Seating assignment available on Blackboard/My Grades
 Please check now, if there are any problems, email me ASAP

- Final Exam Monday 23 May
- Seating assignment available on Blackboard/My Grades
 Please check now, if there are any problems, email me ASAP
- Next Tuesday 17 May, we will have a Mock Exam

- Final Exam Monday 23 May
- Seating assignment available on Blackboard/My Grades
 Please check now, if there are any problems, email me ASAP
- Next Tuesday 17 May, we will have a Mock Exam
 - ► Room 118 Hunter North (Assembly Hall), ground floor of the North Building

- Final Exam Monday 23 May
- Seating assignment available on Blackboard/My Grades
 Please check now, if there are any problems, email me ASAP
- Next Tuesday 17 May, we will have a Mock Exam
 - Room 118 Hunter North (Assembly Hall), ground floor of the North Building
 - ▶ Only 1.15 hours for the Mock, 2 hours for the real exam.

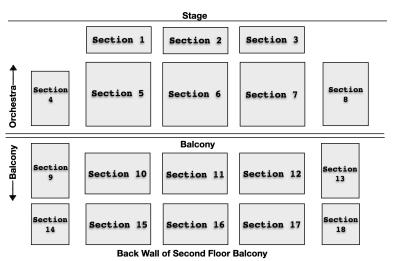
- Final Exam Monday 23 May
- Seating assignment available on Blackboard/My Grades
 Please check now, if there are any problems, email me ASAP
- Next Tuesday 17 May, we will have a Mock Exam
 - Room 118 Hunter North (Assembly Hall), ground floor of the North Building
 - ▶ Only 1.15 hours for the Mock, 2 hours for the real exam.
 - ▶ Just a practice run, this WILL NOT be the same as the real exam, and it will not be graded.

3 / 43

CSci 127: Introduction to Computer Science

CSci 127 Final Exam

Seating alphabetical in each section. See Blackboard for assigned seating section.



What's the best way to study for the final exam?

• 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.

• 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.

- 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 Pass/NoCredit?

- 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 Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study.

- 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 Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study.

Deadline: Friday, April 15, 2022, 11:59 PM (EST)

- 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 h.
 - 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 Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
 - Deadline: Friday, April 15, 2022, 11:59 PM (EST)
- Why do you care about cheating?

- 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 he
 - 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 Pass/NoCredit?
 It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
 - Deadline: Friday, April 15, 2022, 11:59 PM (EST)
- Why do you care about cheating?
 First: it gives unfair advantage & is immoral.

- 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.
 Post properties of the previews are the problems.
 - 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 Pass/NoCredit?
 It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
 - Deadline: Friday, April 15, 2022, 11:59 PM (EST)
- Why do you care about cheating?

 First: it gives unfair advantage & is imm
 - First: it gives unfair advantage & is immoral.
 - Second: it degrades the quality of our students.

5 / 43

CSci 127 (Hunter) Lecture 13 10 May 2022

• 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 Pass/NoCredit?
 It's fine with us, but check with your advisor to make sure it's accepted for your program of study.

Deadline: Friday, April 15, 2022, 11:59 PM (EST)

• Why do you care about cheating?

First: it gives unfair advantage & is immoral.

Second: it degrades the quality of our students.

Third: it's a standard question on faculty references.

10 May 2022

5 / 43

• 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 Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study.

Deadline: Friday, April 15, 2022, 11:59 PM (EST)

• Why do you care about cheating?

First: it gives unfair advantage & is immoral.

Second: it degrades the quality of our students.

Third: it's a standard question on faculty references.

Industry & graduate schools hate it: don't want someone who falsifies work.

5 / 43

CSci 127 (Hunter) Lecture 13 10 May 2022

Today's Topics

```
//Amother (** program, demonstrating I/O & arithmetic finctude closterous using namespote std; int main () {
flood kg. lbs; out = "there kg:"; cin > kg. lbs = kg * 2.2; cat < "there kg. lbs: " < lbs < "\n\n"; return 0; return 0; return 0; return 0;
```

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

Today's Topics

```
//Acother (-* program, demonstrating L/O & arithmetic
finitude dostration outsign amespace std;
int main O
{ foot kg. Ubs;
cost < "force kg:";
cost < "force kg:"
```

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

CSci 127 (Hunter)

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

Efficient for systems programming.

```
//hother C++ program, demonstrating L/0 & arithmetic sinclude sisterneon using nonespace std; int main O { float k_B _lbs; float k_B _lbs; can > k_B _lbs = k_B *2.2; cout <= end! <= "Lbs: " << "\n\n"; return 0; }
```

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

```
//Another C++ program, demonstrating I/O & arithmetic finclude clostream. using namespace std; int moin C) { floot kg, lbs; cout <= fifther kg: "; cin >> kg; lbs = kg * 2.2; cout <= end! <= 'Lbs: " << lbs <= "\n\n"; return 0; } }
```

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

```
//Another C++ program, demonstrating I/O & arithmetic finitude clostreams using namespace std; 
int main O {
    float kg, lbs; 
    cout << "Enter kg: "; 
    lbs = kg." 2.2; 
    cout << center cout << "this: " << lbs << "\n\n"; 
    return 0; 
    return 0; 
    return 0; 
}
```

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

```
//Another C++ program, demonstrating I/O & arithmetic finiclude clostream-
using namespace std;
int main () {
    floot kg, lbs;
    cout <= "Enter kg: ";
    cin >> kg;
    lbs = kg * 2.2;
    cout <= end! <= "Lbs: " << lbs <= "\n\n";
    return 0;
}
```

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

```
//Another C++ program, demonstrating I/O & arithmetic finitude clostreams using namespace std; 
int main O {
    float kg, lbs; 
    cout <= Enter kg; "; 
    in >> kg; 
    lbs = kg * 2.2; 
    cout <= entl <= "ths: " << 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, ...

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

```
//Another (++ program, demonstrating I/O & arithmetic
sinclude <lastream
using namespace std;
int main ()
{
    float kg, lbs;
    cout << "Enter kg: ";
    cin > kg;
    classes, 2,2;
    cout << endl << "Lbs: " << lbs << "\n\n";
    return 0;
}</pre>
```

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

```
//Monther C++ program, demonstrating I/O & arithmetic
finclude cistream
using namespace std;
int main O {
   float kg, lbs;
   cout << "Enter kg: ";
   cin >> kg;
   lbs = kg " 2.2;
   cout << endl << "Lbs: " << lbs << "\n\n";
   return 0;</pre>
```

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

- 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;

```
//Another C++ program, demonstrating I/O & orithmetic finitude clostreams using nomespace std; int motin () {
    floot kg, lbs; cout << "Enter kg: "; ctn >> kg; lbs << "\n\n"; return 0; lbs: " << lbs << "\n\n"; return 0; return 0; return 0; lbs: " << lbs << "\n\n"; return 0; lbs: " << lbs: "\n\n"; return 0; 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:

```
//Another C++ program, demonstrating I/O & arithmetic #include <lostream using nomespace sdd; int main () { float kg, lbs; cut << "Enter kg: "; cut << "Enter kg: "; cut > kg; cl; cot << "enter kg: "; cut > kg; cl; cot << enter kg: "; cut > kg; cl; cot << enter kg: " << 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;

```
//Another (++ program, demonstrating I/O & arithmetic minclude <iostream using namespace sdf; int main O { float kg, lbs; cout << "Enter kg: "; cout << "Enter kg: "; cout << "total ba = kg = 2.2; cout << endl << "lbs: " << lbs << "whin"; 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:

```
//Another C++ program, demonstrating I/O & arithmetic finclude <lostream using namespace std; int main () { floot kg, lbs; cout << "Enter kg: "; cti >> kg; lb; cout << "enter kg: "; cti >> kg; lb; cout << enter kg: "; cti >> kg; lb; cout << enter kg: "</pre>
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: int num;
- Many types available: int, float, char, ...
- To print: cout << "Hello!!";
- To get input: cin >> num;
- To use those I/O functions: #include <iostream> using namespace std;
- Definite loops:
 for (i = 0; i < 10; i++) {...}</pre>

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;
int main ()
 float ka. lbs:
 cout << "Enter ka: ":
 cin >> ka:
 lbs = ka * 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 '}'.

```
//Another C++ program, demonstrating I/O & arithmetic #include <iostreams using nomespace std; tint 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++) {...}</pre>
- Blocks of code uses '{' and '}'.
- Commands generally end in ';'.

Today's Topics

```
//Acother (** program, demonstrating I/O & arithmetic Binclude closterous total grammapure Std; int main ()
{
floot kg, lbs; coot ** "Inter kg; "; coot ** "There kg; "; coot ** "There kg; "; coot ** "Ener kg; "; coot ** "Coot ** "Inter kg; "; coot ** "Red ** "Lbs; " ** lbs ** "Vn'n"; "etcum 6; "
return 6; "
```

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

CSci 127 (Hunter)

Challenge:

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";</pre>
    else if (yearBorn <= 1964)
        cout << "Baby Boomer":
    else if (yearBorn <= 1984)
        cout << "Generation X";</pre>
    else if (vearBorn <= 2004)
        cout << "Millennial":</pre>
    else
        cout << "TBD":
    return 0:
   CSci 127 (Hunter)
```

```
using namespace std;
     int main ()
         string conditions = "blowing snow";
         int winds = 100;
         float visibility = 0.2;
         if ( ( (winds > 35) && (visibility < 0.25) )
               ( (conditions == "blowing snow") ||
                 (conditions == "heavy snow") ) )
             cout << "Blizzard!\n":</pre>
         string origin = "South Pacific";
         if (winds > 74)
             cout << "Major storm, called a ";</pre>
         if ((origin == "Indian Ocean")
             |/(origin == "South Pacific"))
             cout << "cyclone.\n";</pre>
         else if (origin == "North Pacific")
             cout << "typhoon.\n";</pre>
         else
             cout << "hurricane.\n";</pre>
                    4 D > 4 A > 4 B > 4 B >
                                                  90 Q
Lecture 13
                                    10 May 2022
                                                   11 / 43
```

C++ Demo

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

Conditionals

General format:

```
//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;
```

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

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

CSci 127 (Hunter) Lecture 13 10 May 2022 14 / 43

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

and (&&)

| in1 | | in2 | returns: |
|-------|----|-------|----------|
| False | && | False | False |
| False | && | True | False |
| True | && | False | False |
| True | && | True | True |

CSci 127 (Hunter) Lecture 13 10 May 2022 14 / 43

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 | 11 | False | False |
| False | 11 | True | True |
| True | \Box | False | True |
| True | 11 | True | True |

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

and (&&)

| in1 | | in2 | returns: |
|-------|----|-------|----------|
| False | && | False | False |
| False | && | True | False |
| True | && | False | False |
| True | && | True | True |
| | | | 1 |

or (||)

| in1 | | in2 | returns: |
|-------|-------|-------|----------|
| False | | False | False |
| False | 11 | True | True |
| True | 11 | False | True |
| True | Π | True | True |

not (!)

| | in1 | returns: |
|---|-------|----------|
| ! | False | True |
| ! | True | False |

Lecture Slip

• Write a C++ program that will ask for the time in 24 hour format and, knowing it is morning before 12pm and evening after 6pm (18), it will print out "Morning", "Afternoon" or "Evening".

CSci 127 (Hunter) Lecture 13 10 May 2022 15 / 43

Today's Topics

```
//Acother C+program, demonstrating I/O & arithmetic finctude -distress continuing namespace std; internation of floor than the std of the std o
```

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

Challenge:

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";</pre>
  while(population < 1000)</pre>
    cout << year << "\t\t" << population << "\n";
    population = population * 2;
    year++;
  return 0;
```

C++ Demo

```
///white Growth Example
#include ciostream>
using namespace std;
int main ()
{
   int population = 100;
   int year = 0;
   cout << "Year'tPopulation\n";
   white(population < 1000)
   {
      cout << year << "\t\t" << population << "\n";
      population = population * 2;
      year++;
   }
   return 0;
}</pre>
```

(Demo with onlinegdb)

Indefinite Loops: while

```
///While Growth Example
#include <iostream>
using namespace std;
int main ()
{
   int population = 100;
   int year = 0;
   cout << "ear\Population\n";
   while(population < 1000)
{
   cout << year << "\t\t" << population << "\n";
   population = population * 2;
   year++;
}
return 0;
}</pre>
```

```
General format:
while ( logical expression )
{
    command1;
    command2;
    command3;
    ...
}
```

Challenge:

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";</pre>
      cout << "Enter an even number: ";</pre>
      cin >> num;
  cout << "You entered: "</pre>
       << num << ".\n";
  return 0:
```

C++ Demo

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

(Demo with onlinegdb)

CSci 127 (Hunter)

Indefinite Loops: while

```
General format:
while ( logical expression )
{
    command1;
    command2;
    command3;
    ...
}
```

Challenge:

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: ";</pre>
      cin >> num;
  } while (num % 2 != 0);
  cout << "You entered: "
       << num << ".\n";
  return 0;
```

10 May 2022

C++ Demo

24 / 43

CSci 127 (Hunter) Lecture 13 10 May 2022

Indefinite Loops: do-while

```
General format:

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

Today's Topics

```
//Another (** program, demonstrating L/O & arithmetic filmiculed constraints of the const
```

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

CSci 127 (Hunter)

```
//Another C+* programs; Demonstrates Loops
###Include clostrones std;
int main O;
int i,i;
for (1 = 0; i < 4; i+*)
{
    cout << "The world turned upside down...\n";
}
for (3 = 10; j > 0; j--)
{
    cout << y <= ";
    cout << "Files off!!" << end1;
    return 0;
}</pre>
```

CSci 127 (Hunter) Lecture 13 10 May 2022 27 / 43

I/O: cin >> ...;

```
//Another C++ program; Demonstrates loops
###Clude clostrong
uning nomespace all;
int main O;
for (1 = 0; i < 4; i++)
{
    cout <<"The world turned upside down...\n";
}
for (3 = 10; j = 0; j--)
{
    cout << "Blast off!!" << end!;
    return 0;
}</pre>
```

CSci 127 (Hunter) Lecture 13 10 May 2022 27 / 43

• I/O: cin >> ...; & cout << ...;

CSci 127 (Hunter) Lecture 13 10 May 2022 27 / 43

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

```
//Another C++ program; Demonstrates loops
###Include clostrates
using nomespace std;
int main O;
for (1 = 0; i < 4; i++)
{
    cout << "The world turned upside down...\n";
}
for (3 = 10; j > 0; j--)
{
    cout << "Tile world turned upside down...\n";
}
cout << "Tile world turned upside down...\n";
}
for (3 = 10; j > 0; j--)
{
    cout << "Tile world turned upside down...\n";
}
return 0;
}</pre>
```

27 / 43

CSci 127 (Hunter) Lecture 13 10 May 2022

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

```
#include clostream
using nomespore std;

int main ()

{
    icn i, j;
    icn i, j;
    cout «~ The world turned upside down...\n';
    for (j = 10; j > 0; j --)
    {
        cout « j « ~ ";
    }
    cout « fells to ff!!" « endl;
    return 0;
```

//Another C++ program; Demonstrates loops

//Another C++ program; Demonstrates loops #include <iostreamusing namespace std;

```
int main O
{
    int i,j
    for G = 0; i < 4; i+*)
{
        cout << "The world turned upside down...\n";
}
    for G = 10; j > 0; j-*)
{
        cout << j << "";
}
        cout << j << "";
}
        cout << j << "";
}
```

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

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

Indefinite loops:

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

for (j = 10; j > 0; j--) { | cout << j << " ":

cout << "Blast off!!" << endl;
return 0;</pre>

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

int main ()
{
 int i,j;
 for (i = 0; i < 4; i++)

Recap: C++ Control Structures

```
• I/O: cin >> ...; & cout << ...;
Definite loops:
  for (i = 0; i < 10; i++)
  Conditionals:
  if (logical expression)
  else
• Indefinite loops:
  while (logical expression)
                      4 D > 4 A > 4 B > 4 B > -
```

int main O
{
 int i, j;
 int i, j;

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

• 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;
}</pre>
```

• Rewrite this program in C++:

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

CSci 127 (Hunter) Lecture 13 10 May 2022 29 / 43

Prewrite this program in C++:

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

#include <iostream>
using namespace std;

CSci 127 (Hunter) Lecture 13 10 May 2022 29 / 43

Print("Year is", i)

#include <iostream>
using namespace std;

 Rewrite this program in C++:

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

#include <iostream>
using namespace std;

int main()

Print("Year is", i)

#include <iostream>
using namespace std;
int main()
Print("Year is", i)

• 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)
```

29 / 43

CSci 127 (Hunter) Lecture 13 10 May 2022

• 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;</pre>
```

• 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;</pre>
  return 0;
```

• Rewrite this program in Python:

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

CSci 127 (Hunter) Lecture 13 10 May 2022 30 / 43

• Rewrite this program in Python:

```
#include <iostream>
using namespace std;
int main()
  for (int i = 1; i < 50; i++)
    cout << i << endl;</pre>
 return 0;
for i in range(1, 50):
```

• Rewrite this program in Python:

```
#include <iostream>
using namespace std;
int main()
  for (int i = 1; i < 50; i++)
    cout << i << endl:</pre>
 return 0;
for i in range(1, 50):
    print(i)
```

30 / 43

```
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).

```
Python: what is the output?
year = 2016
if year % 4 == 0 and \
    (not (year % 100 == 0) or (year % 400 == 0)):
    print("Leap!!")
print("Year")
```

CSci 127 (Hunter) Lecture 13 10 May 2022 32 / 43

```
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")
```

32 / 43

```
Python: what is the output?
year = 2016
if year % 4 == 0 and \
    (not (year % 100 == 0) or (year % 400 == 0)):
    print("Leap!!")
print("Year")
```

CSci 127 (Hunter) Lecture 13 10 May 2022 33 / 43

```
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")
```

```
Python: what is the output?
year = 2016
if year % 4 == 0 and \
    (not (year % 100 == 0) or (year % 400 == 0)):
    print("Leap!!")
print("Year")
```

CSci 127 (Hunter) Lecture 13 10 May 2022 34 / 43

```
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")
```

```
Python: what is the output?
year = 2016
if year % 4 == 0 and \
    (not (year % 100 == 0) or (year % 400 == 0)):
    print("Leap!!")
print("Year")
```

CSci 127 (Hunter) Lecture 13 10 May 2022 35 / 43

```
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")
```

```
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")
```

```
Python: what is the output?
  year = 2016
  if year % 4 == 0 and \setminus
      (not (year \% 100 == 0) or (year \% 400 == 0)):
      print("Leap!!")
  print("Year")
  year = 2016
  if TRUE and \
      (TRUE):
      print("Leap!!")
  print("Year")
```

```
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")
```

38 / 43

```
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

• 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;
```

CSci 127 (Hunter) Lecture 13 10 May 2022 39 / 43

• 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()
```

CSci 127 (Hunter) Lecture 13 10 May 2022 39 / 43

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

• 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:";</pre>
```

CSci 127 (Hunter) Lecture 13 10 May 2022 39 / 43

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

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

```
#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";
   }</pre>
```

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

• 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:";</pre>
  cin >> rides;
  if (2.75 * rides < 33.00)
    cout << "Cheaper to buy single ride metro cards.\n";</pre>
  else
    cout << "Cheaper to buy 7-day unlimited card.\n";</pre>
```

39 / 43

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

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

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

CSci 127 (Hunter) Lecture 13 10 May 2022 40 / 43

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

<ロト < @ ト < き > < き > き り < で

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

```
s = ""
```

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

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

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

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

• 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 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.

• 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;
```

• 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()
```

• 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;
```

• 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)
```

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

• 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)
```

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

• 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)
```

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

• 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)
```

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

• 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:";</pre>
    cin >> num;
  return 0;
```

10 May 2022



Before next lecture, don't forget to:

Work on this week's Online Lab



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001E Hunter North



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001E Hunter North
- Submit this week's 5 programming assignments (programs 57-60)



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001E Hunter North
- Submit this week's 5 programming assignments (programs 57-60)
- If you need help, schedule an appointment for Tutoring in lab 1001E 11am-5pm



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001E Hunter North
- Submit this week's 5 programming assignments (programs 57-60)
- If you need help, schedule an appointment for Tutoring in lab 1001E 11am-5pm
- Take the Lecture Preview on Blackboard on Monday (or no later than 10am on Tuesday)

Lecture Slips & Writing Boards



- Hand your lecture slip to a UTA.
- Return writing boards as you leave.