CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

This lecture will be recorded

CSci 127 (Hunter) Lecture 13 1 Dec

• Final Exam Monday 14 December

- Final Exam Monday 14 December
- Deadline for choosing Early exam is on December 4
 Submit Final Exam Date Choice on Gradescope
 If you don't submit, we will assume you are taking the exam on 14
 December.

CSci 127 (Hunter) Lecture 13

- Final Exam Monday 14 December
- Deadline for choosing Early exam is on December 4
 Submit Final Exam Date Choice on Gradescope
 If you don't submit, we will assume you are taking the exam on 14
 December.
- NextTuesday 8 December we will have a Mock Exam

CSci 127 (Hunter) Lecture 13

- Final Exam Monday 14 December
- Deadline for choosing Early exam is on December 4
 Submit Final Exam Date Choice on Gradescope
 If you don't submit, we will assume you are taking the exam on 14
 December.
- NextTuesday 8 December we will have a Mock Exam
 - ► Come to lecture, login to Gradescope and take the exam.

- Final Exam Monday 14 December
- Deadline for choosing Early exam is on December 4 Submit Final Exam Date Choice on Gradescope If you don't submit, we will assume you are taking the exam on 14 December.
- NextTuesday 8 December we will have a Mock Exam
 - ► Come to lecture, login to Gradescope and take the exam.
 - ▶ I will be available on chat to answer questions.

2/39

CSci 127 (Hunter) Lecture 13

- Final Exam Monday 14 December
- Deadline for choosing Early exam is on December 4 Submit Final Exam Date Choice on Gradescope If you don't submit, we will assume you are taking the exam on 14 December.
- NextTuesday 8 December we will have a Mock Exam
 - ► Come to lecture, login to Gradescope and take the exam.
 - ▶ I will be available on chat to answer questions.
 - ▶ Only 1.15 hours for the Mock, 2 hours for the real exam.

2/39

CSci 127 (Hunter) Lecture 13

- Final Exam Monday 14 December
- Deadline for choosing Early exam is on December 4
 Submit Final Exam Date Choice on Gradescope
 If you don't submit, we will assume you are taking the exam on 14
 December.
- NextTuesday 8 December we will have a Mock Exam
 - ► Come to lecture, login to Gradescope and take the exam.
 - ▶ I will be available on chat to answer questions.
 - ▶ 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.

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 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 and work through, grade yourself, update note sheet, and repeat.
- 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 hour

and work through, grade yourself, update note sheet, and repeat.

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. Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour
- and work through, grade yourself, update note sheet, and repeat.
 Why do you care about cheating?
 First, it gives upfair advantage for its immoral.
 - First: it gives unfair advantage & is immoral. Second: it degrades the quality of our students.

• 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

and work through, grade yourself, update note sheet, and repeat.

• 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.

• 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 and work through, grade yourself, update note sheet, and repeat.

• 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.

Today's Topics

```
//Acother (-- program, demonstrating I/O & arithmetic finitude closterous targ namespace std; int main () { floot kg, lbs; on a force kg; "; lbs - kg *2.2; cat <- mail <- mai
```

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

Today's Topics

```
//Acother C+p program, demonstrating I/O & arithmetic finitude closterose std; int main () { | floot kg, lbs; coat < 'Erier kg, ''; coat < 'Erier kg, '';
```

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

```
//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.

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

1 December 2020

7/39

CSci 127 (Hunter) Lecture 13

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

```
//Another C++ program, demonstrating L/0 & arithmetic finclude clostreams using namespace std; int main O { { floot kg, lbs; cout <= "Enther kg: "; cit >> kg; lbs = kg * 2.2; cout <= endl <= "Lbs: " << lbs << "\n\n"; return 0; } }
```

7/39

CSci 127 (Hunter) Lecture 13 1 December 2020

- 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 ka: ":
 cin >> ka:
 lbs = ka * 2.2;
 cout << endl << "Lbs: " << lbs << "\n\n";
 return 0;
```

7/39

CSci 127 (Hunter) Lecture 13

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

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

CSci 127 (Hunter) Lecture 13

- 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 clostream-
using namespace std;

int main () {
    float kg, lbs;
    cout << "Enter kg;";
    cin >> 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, ...

```
//Another C++ program, demonstrating I/O & arithmetic
finclude cistream
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;</pre>
```

- 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 finclude <lastream using namespace std; 
int main O {
    float kg, lbs; 
    cut << "Enter kg: "; 
    cin >> kg; 
    loat kg = 2.2; 
    cout << end << "Lbs: " << lbs << "\n\n";
```

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

```
//Another C++ program, demonstrating I/O & arithmetic
finclude <isotremn
using namespace std;
int main O {
    float kg, lbs;
    cout << "Enter kg: ";
    cin >> kg;
    lbs = kg * 2.2;
    cout << end! << "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;

- 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 (++ program, demonstrating I/O & arithmetic minclude <lastream using namespace std;
int main O {
float kg, lbs;
cout << Enter kg: ";
cin > kg;
cin > kg;
cout << et al. 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: int, float, char, ...
- To print: cout << "Hello!!";
- To get input: cin >> num;
- To use those I/O functions:
 #include <iostream>
 using namespace std;

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

```
• 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>

```
    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 '}'.

```
//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 '}'.
- Commands generally end in ';'.

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

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 >
Lecture 13
                                                   9/39
```

1 December 2020

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 1 December 2020 12 / 39

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 1 December 2020 12 / 39

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	\Box	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
			•

in1		in2	returns:
False		False	False
False	Π	True	True
True	\Box	False	True
True	Π	True	True

or (||)

not (!)

in1		returns:	
!	False	True	
!	True	False	

Lecture Quiz

- Log-in to Gradescope
- Find LECTURE 13 Quiz
- Take the quiz
- You have 3 minutes

CSci 127 (Hunter) Lecture 13 1 December 2020 13 / 39

Today's Topics

```
//Acother C++ program, demonstrating L/O & arithmetic finitudes dostrated using namespace std; int main () { floot kg, lbs; cost < 'Uner kg, 't'; lbs < 'g' 2.2; cost < endi of "lbs: 'kg '2.2; cost < endi of "lbs: 'kg '2.2; cost < endi of "lbs: 'kg '2.8; 'return 0; return 0;
```

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

(Demo with onlinegdb)

CSci 127 (Hunter) Lecture 13 1 December 2020 16 / 39

Indefinite Loops: while

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

int main ()
{
   int population = 100;
   int year = 0;
   cout << "Year\Population\n";
   while (population < 1000)
{
     cout << year << "\t" << population << "\n";
     population = population * 2;
   }
   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 != \emptyset)
      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) Lecture 13 1 December 2020 19 / 39

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;
```

1 December 2020

C++ Demo

Indefinite Loops: do-while

```
General format:

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

Today's Topics

```
//Another C+ program, demonstrating I/O & arithmetic finitudes dostrate using namespace std; iten sain () {
floot kg, lbs; cout « 'Erber kg: ", cout « 'Erber kg: ", lbs » kg " 2.2; cout « endi « "lbs: " « lbs « "\n'n"; return 0;
```

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

CSci 127 (Hunter)

CSci 127 (Hunter) Lecture 13 1 December 2020 25 / 39

• I/O: cin >> ...;

```
//Another C++ program; Demonstrates loops
disclude -distrete cut program; Demonstrates loops
disclude -distrete cut program;
for (j = 0; i < 4; i+>)
for (j = 10; j < 0; j--)
cut < c' j < c' '';
}
cut < c' j < "'';
}
cut < c' j < "'';
return 0;
}
```

CSci 127 (Hunter) Lecture 13 1 December 2020 25 / 39

• I/O: cin >> ...; & cout << ...;

```
//Another C++ programs; Demonstrates loops flictuale clostresponding namespace std; int sin; G int sin; G int G interpretable G int
```

CSci 127 (Hunter) Lecture 13 1 December 2020 25 / 39

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

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

25/39

CSci 127 (Hunter) Lecture 13 1 December 2020

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

```
//Another (++ program; Demonstrates loops
Binclude -dostree std;
int main ()
{
   int i, j;
   for (j = 0; i < 4; i++)
   {
        cout < "The world turned upside down...\n";
   }
   for (j = 10; j > 0; j--)
   {
        cout < c j < c ";
   }
   cout < c "Blost offil" << end1;
   return 0;
}</pre>
```

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

```
int main ()  \begin{cases} \text{int } i,j \\ \text{int } i,j \\ \text{for } (i=0;\ i<4;\ i++) \\ \text{cout} << \text{'The world turned upside down...\n'';} \\ \text{for } (j=10;\ j>0;\ j--) \\ \text{cout} << j << \text{''};} \\ \text{cout} << \text{'Blast off!}!' << \text{end1}; \\ \text{return 0};
```

Conditionals:

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

```
\label{eq:continuous} / \text{//Another (+) program; Demonstrates loops} \\ \text{Minclude -dostrone} \\ \text{with minclude of the minclu
```

```
• 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++)

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

int main ()
{
 int i,j;
 for (; 0; 1 < 4; 1++)
}
cout << "line i, 1 < 0; 1 < 4; 1++)
{
 cout << "The world turned upside down...\n";
}
cout << "line i, j < 0; j --)
{
 cout << "Blast off!!" << end1;
}
return 0;
}

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

4 D b 4 A B b 4 B b B

• 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 1 December 2020 27 / 39

Prewrite this program in C++:

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

#include <iostream>
using namespace std;

27 / 39

CSci 127 (Hunter) Lecture 13 1 December 2020

Print("Year is", i)

#include <iostream>
using namespace std;

using namespace std;

Print("Year is", i)

#include <iostream>
using namespace std;

int main()

Print("Year is", i)

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

Rewrite this program in C++:

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

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

• 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)
```

27 / 39

CSci 127 (Hunter) Lecture 13 1 December 2020

• 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>
```

CSci 127 (Hunter)

• 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;
```

27 / 39

• 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)

• 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)
```

28 / 39

```
• 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")
```

```
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")
```

```
Python: what is the output?
year = 2016
if year % 4 == 0 and \
    (not (year % 100 == 0) 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")
  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")
```

32 / 39

```
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")
```

```
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 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 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")
```

```
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;
```

• 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()
```

```
#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>
```

 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;
```

37 / 39

• 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)</pre>
```

37 / 39

```
#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
```

```
#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>
```

CSci 127 (Hunter)

• 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>
  return 0;
```

Lecture 13

1 December 2020

37 / 39

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

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

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

• 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()
```

39 / 39

• 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;
```

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

```
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.

Lecture 13

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

CSci 127 (Hunter)

1 December 2020

39 / 39



Before next lecture, don't forget to:

Work on this week's Online Lab

40 / 39



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Optional attend live Lab Review on Wednesday 1-2:30pm

40 / 39



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Optional attend live Lab Review on Wednesday 1-2:30pm
- Take the Lab Quiz on Gradescope by 6pm on Wednesday

40 / 39



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Optional attend live Lab Review on Wednesday 1-2:30pm
- Take the Lab Quiz on Gradescope by 6pm on Wednesday
- Submit this week's 4 programming assignments (programs 57-60)

40 / 39



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Optional attend live Lab Review on Wednesday 1-2:30pm
- Take the Lab Quiz on Gradescope by 6pm on Wednesday
- Submit this week's 4 programming assignments (programs 57-60)
- At any point, visit our Drop-In Tutoring 11am-5pm for help!!!



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Optional attend live Lab Review on Wednesday 1-2:30pm
- Take the Lab Quiz on Gradescope by 6pm on Wednesday
- Submit this week's 4 programming assignments (programs 57-60)
- At any point, visit our Drop-In Tutoring 11am-5pm for help!!!
- Take the Lecture Preview on Blackboard on Monday (or no later than 10am on Tuesday)