CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

990

1/40

This lecture will be recorded

CSci 127 (Hunter) Lecture 13 4 May 2021

Final Exam Monday 24 May

- Final Exam Monday 24 May
- Deadline for choosing Early exam is on May 10
 Submit Early Final Exam Option on Gradescope

- Final Exam Monday 24 May
- Deadline for choosing Early exam is on May 10
 Submit Early Final Exam Option on Gradescope
 If you don't submit, we will assume you are taking the exam on 24 May.

- Final Exam Monday 24 May
- Deadline for choosing Early exam is on May 10
 Submit Early Final Exam Option on Gradescope
 If you don't submit, we will assume you are taking the exam on 24 May.
 - If you choose the Early Exam, you CANNOT take the exam on 24 May.

- Final Exam Monday 24 May
- Deadline for choosing Early exam is on May 10
 Submit Early Final Exam Option on Gradescope
 If you don't submit, we will assume you are taking the exam on 24 May.
 - If you choose the Early Exam, you CANNOT take the exam on 24 May.
- NextTuesday 11 May we will have a Mock Exam

- Final Exam Monday 24 May
- Deadline for choosing Early exam is on May 10
 Submit Early Final Exam Option on Gradescope
 If you don't submit, we will assume you are taking the exam on 24 May.
 - If you choose the Early Exam, you CANNOT take the exam on 24 May.
- NextTuesday 11 May we will have a Mock Exam
 - ► Come to lecture, login to Gradescope and take the exam.

- Final Exam Monday 24 May
- Deadline for choosing Early exam is on May 10
 Submit Early Final Exam Option on Gradescope
 If you don't submit, we will assume you are taking the exam on 24 May.
 - If you choose the Early Exam, you CANNOT take the exam on 24 May.
- NextTuesday 11 May 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.

- Final Exam Monday 24 May
- Deadline for choosing Early exam is on May 10
 Submit Early Final Exam Option on Gradescope
 If you don't submit, we will assume you are taking the exam on 24 May.
 - If you choose the Early Exam, you CANNOT take the exam on 24 May.
- NextTuesday 11 May 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.

4 May 2021

2 / 40

CSci 127 (Hunter) Lecture 13

- Final Exam Monday 24 May
- Deadline for choosing Early exam is on May 10
 Submit Early Final Exam Option on Gradescope
 If you don't submit, we will assume you are taking the exam on 24 May.
 - If you choose the Early Exam, you CANNOT take the exam on 24 May.
- NextTuesday 11 May 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.

CSci 127 (Hunter) Lecture 13

2 / 40

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
- Why do you care about cheating?
 First: it gives unfair advantage & is immoral.
 Second: it degrades the quality of our students.

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

3 / 40

CSci 127 (Hunter) Lecture 13 4 May 2021

• 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 how.

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.

4 May 2021

3 / 40

• 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.

3 / 40

CSci 127 (Hunter) Lecture 13 4 May 2021

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

```
//Another C+ program, demonstrating I/O & arithmetic finctude clasters using namespace std; int main () {
floot kg, lbs; cost = "faster kg; "; cost = "fas
```

- 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 minclude ciastream using namespose st;
int main () {
    floot 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.

```
//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; } }
```

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

```
//Another C++ program, demonstrating I/O & arithmetic finiclude clostream-
using namespace std;
int main O {
    float kg, lbs;
    cout <= Enter kg: ";
    in >= 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;

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

- 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, ...

```
//Another (++ program, demonstrating I/O & arithmetic
finclude <lastrama
using namespace std;
int main ()
{
    float kg, lbs;
    cout << "Enter kg: ";
    cin >> kg;
    lbs = kg " 2.2;
    cout << entl << "lbs: " << lbs << "\n\n";
    return 6;</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
sinclude <lastream
using namespace std;
int main ()
{
    float kg, lbs;
    cout << "Enter kg: ";
    cin > kg;
    cout << et al. 2, 2;
    cout << et al. 2, 2;
    cout << et al. 2, 2;
    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!!";

```
//Another 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;

- 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 finclude <lastream suing namespace std; int main O { float kg, lbs; cout <= Enter kg: "; cin >> kg; cin >> kg; 2; lout <= 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!!";
- 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>

```
//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 '}'.

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

```
//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 (-s program, demonstrating I/O & arithmetic finitude closterous tag' intension ()

{floot kg. lbs;
coat < fineth kg. 'lbs;
cia > kg' 2.2;
coat < end < "lbs < kg "2.2;
coat < end < "lbs < "lbs < "unin";
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:

```
//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 >
```

4 May 2021

9 / 40

Lecture 13

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:

```
if ( logical expression )
//Demonstrates conditionals
#include <iostream>
using namespace std;
                                                command1;
int main ()
   int yearBorn:
                                                . . .
   cout << "Enter year born: ";
   cin >> yearBorn;
   if (yearBorn < 1946)
      cout << "Greatest Generation";
                                      else if ( logical expression )
   else if (yearBorn <= 1964)
      cout << "Baby Boomer";
   else if (yearBorn <= 1984)
                                                command1;
      cout << "Generation X":
   else if (yearBorn <= 2004)
      cout << "Millennial":
   else
                                      else
      cout << "TBD":
   return 0;
                                                command1;
```

...

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

CSci 127 (Hunter) Lecture 13 4 May 2021 12 / 40

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 4 May 2021 12 / 40

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

in1		in2	returns:
False		False	False
False	11	True	True
True	11	False	True
True	-11	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 4 May 2021 13 / 40

Today's Topics

```
//Another C+program, demonstrating I/O & arithmetic Binclude closterous using namespace std; int main O {
floot kg, lbs; coat ~ "Inter kg:";
lbs = kg " 2.2;
coat ~ end! ~ "lbs: " ~ lbs ~ "whn";
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)

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)

19 / 40

Indefinite Loops: while

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

20 / 40

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;
```

4 May 2021

C++ Demo

Indefinite Loops: do-while

```
General format:

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

Today's Topics

```
//Amother C.s program, demonstrating I/O & arithmetic diriculae closterous using nomespace std; it nein () { floot kg, lbs; coat « "Erber kg; "; coat « "Erber kg; "; lbs » kg " 2.2; coat « end! « "Lbs: " « lbs « "Whin"; return 0;
```

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

• I/O: cin >> ...;

```
//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>
```

4 May 2021

25 / 40

CSci 127 (Hunter) Lecture 13

• I/O: cin >> ...; & cout << ...;

CSci 127 (Hunter)

- 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>
```

```
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 <iostreamusing namespace std;

```
Conditionals:
```

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

```
//Another (++ program) Demonstrates loops
Minimum another that it was to limit with 0

{
    int i, 1;
    for (' = 0; i < 4; i++)
    {
        cout << "The world turned upside down...\n";
    }
    for (5 = 10; j > 0; j--)
    {
        cout << 'Bast off!!" << end!;
    return 0;
}
```

```
• 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++)

25 / 40

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

//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++)

4 - 1 - 4 - 4 - 5 - 4 - 5 - 5

• 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>
```

26 / 40

Rewrite this program in C++:

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

CSci 127 (Hunter) Lecture 13 4 May 2021 27 / 40

Prewrite this program in C++:

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

#include <iostream>
using namespace std;

CSci 127 (Hunter)

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()

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)
```

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>
```

• 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 / 40

• 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 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 / 40

```
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)

```
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")
```

CSci 127 (Hunter)

```
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)

```
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")
```

33 / 40

CSci 127 (Hunter) Lecture 13 4 May 2021

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

35 / 40

```
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;
```

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

```
#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>
```

37 / 40

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

4 May 2021

37 / 40

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

• 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)
```

```
#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)
```

```
#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)
```

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

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

39 / 40

• 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;
```

4 May 2021



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
- Optional attend Lab Review (Zoom links on Blackboard / Syncrhonous Meetings)



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Optional attend Lab Review (Zoom links on Blackboard / Syncrhonous Meetings)
- Take the Lab Quiz on Gradescope by 6pm on Wednesday



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Optional attend Lab Review (Zoom links on Blackboard / Syncrhonous Meetings)
- Take the Lab Quiz on Gradescope by 6pm on Wednesday
- Submit this week's 4 programming assignments (programs 57-60)



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Optional attend Lab Review (Zoom links on Blackboard / Syncrhonous Meetings)
- 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!!!

4 □ ▶ 4 □ № 4 □ №

CSci 127 (Hunter) Lecture 13 4



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Optional attend Lab Review (Zoom links on Blackboard / Syncrhonous Meetings)
- 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)