CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

• Final Exam next Monday 18 May (in 6 days!)

- Final Exam next Monday 18 May (in 6 days!)
- The day before the final exam make sure you have access to the exam course on Gradescope (single exam assignment) Contact us right away if you don't!

- Final Exam next Monday 18 May (in 6 days!)
- The day before the final exam make sure you have access to the exam course on Gradescope (single exam assignment)
 Contact us right away if you don't!
- No Code Review this week

- Final Exam next Monday 18 May (in 6 days!)
- The day before the final exam make sure you have access to the exam course on Gradescope (single exam assignment)
 Contact us right away if you don't!
- No Code Review this week
- Online help available until Thursday May 14
 - ► Email: csci127help@gmail.com

- Final Exam next Monday 18 May (in 6 days!)
- The day before the final exam make sure you have access to the exam course on Gradescope (single exam assignment)
 Contact us right away if you don't!
- No Code Review this week
- Online help available until Thursday May 14
 - ► Email: csci127help@gmail.com
 - ▶ Discussion Board: on Blackboard, link on purple menu bar

- Final Exam next Monday 18 May (in 6 days!)
- The day before the final exam make sure you have access to the exam course on Gradescope (single exam assignment)
 Contact us right away if you don't!
- No Code Review this week
- Online help available until Thursday May 14
 - ► Email: csci127help@gmail.com
 - Discussion Board: on Blackboard, link on purple menu bar
 - ► Drop-in tutoring (11am-6:30pm):

- Final Exam next Monday 18 May (in 6 days!)
- The day before the final exam make sure you have access to the exam course on Gradescope (single exam assignment)
 Contact us right away if you don't!
- No Code Review this week
- Online help available until Thursday May 14
 - ► Email: csci127help@gmail.com
 - ▶ Discussion Board: on Blackboard, link on purple menu bar
 - ➤ Drop-in tutoring (11am-6:30pm): sign in here: https://bit.ly/csci127Tutoring then join the session here: https://bit.ly/csci127TutoringSession

 \bullet 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.
- 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 unfair advantage & is immoral.
 Second: it degrades the quality of our students.

12 May 2020

• 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

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.

3/39

CSci 127 (Hunter) Lecture 13 12 May 2020

• 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.

3/39

• 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.

3/39

Today's Topics

```
//Amother C+program, demonstrating I/O & arithmetic finitude clostrate using namespace std; int main () { floot kg, lbs; cout < "Errer kg:"; cout < "Errer kg:"; lbs < kg. 2.2; cout < endi or "Lbs: " < lbs < "\whin"; 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

```
//Amother (** program, demonstrating I/O & arithmetic Birclade -dostream using namespack Md; int main ()

{floot kg, lbs; coat ** coat ** coat ** cfiner kg; cia ** kg; lbs ** kg ** 2.2; coat ** cfiner kg ** ".v'un"; return 0;

return 0;
```

- 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
#include clostreams
using namespace std;
int main O
{
float kg, lbs;
cout << "Enter kg: ";
cls > kg;
lbs = kg ' 2.2;
cout << end! << "Lbs: " << lbs << "\n\n";
return 0;
}
}</pre>
```

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

- 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 finiculed cisostremm using namespace std; int main () {
    floot kg, lbs;
    cout < "Enter kg: ";
    cin >> kg;
    lbs = kg * 2.2;
    cout < e mell < "Lbs: " << lbs << "\n\n";
    return 0;
}
```

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

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

```
//honther (++ program, demonstrating I/O & arithmetic finclude <lastrambusing namespace std; 
int main () {
    float kg, lbs; 
    cout << "Enter kg: "; 
    cin >> kg; 
    lbs / kg 2.2; 
    cout << end! << "tbs: " << lbs << "\n\n"; 
    teturn 0;
```

7/39

- 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 <lastrambusing namespace std; 
int main () {
   float kg. 'bs; 
   cot <= "Enter kg: "; 
   lbs = kg * 2.2; 
   cout <= endl <= "Libs: " << 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!!";

```
//Another C++ program, demonstrating I/O & arithmetic
finclude <isotreom
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:

//Another C++ program, demonstrating I/O & arithmetic

- 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 & arithmetic
sinclude <lastrambusing namespace std;
int main O
{
    float kg, lbs;
    cout << "Enter kg: ";
    cin >> kg;
    lbs = kg " 2.2;
    cout << entl << "Lbs: " << lbs << "\n\n";
    return 0;</pre>
```

7/39

- 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: "; cout << enter kg: "!bs: " << 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 finclude <lastream using namespace std; int main () { float kg, lbs; cout << "Enter kg: "; cout << "Enter kg: "; cout <= table = kg * 2.2; cout << entl << "tbs: " << 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, ...
- o To print: cout << "Hello!!";</pre>
- To get input: cin >> num;
- To use those I/O functions: #include <iostream> using namespace std;
- Definite loops:

7/39

```
//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: "..." < 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>

```
//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 <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

```
//Amother (** program, demonstrating I/O & arithmetic fluctual constraint of the fluctual constraint 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:

```
//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 >
```

12 May 2020

9/39

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;
                                               ...
```

11/39

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

CSci 127 (Hunter) Lecture 13 12 May 2020 12 / 39

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

and (&&)

	in2	returns:
&&	False	False
&&	True	False
&&	False	False
&&	True	True
	&& &&	&& False && True && False

CSci 127 (Hunter) Lecture 13 12 May 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
			1

or (||)

in1		in2	returns:
False		False	False
False	11	True	True
True	11	False	True
True	\Box	True	True

not (!)

	in1	returns:
!	False	True
!	True	False

Today's Topics

```
//Acother (** program, demonstrating L/O & arithmetic flintinded-constraint program, and managenes std; int main () {
float kg, lbs; coat ~ 'Enter kg; ";
lbs = kg ' 2.2; coat ~ end < "lbs: "ks ' 2.2; coat ~ end < "lb
```

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

13 / 39

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;
```

CSci 127 (Hunter) Lecture 13 12 May 2020 14 / 39

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 12 May 2020 15/39

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)

Indefinite Loops: while

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

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

19 / 39

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;
```

20 / 39

C++ Demo

```
//Demonstrates do-while loops
#include <iostream>
using namespace std;
int main ()
{
    int num;
    do
    {
        cout << "Enter an even number: ";
        cin >> num;
} while (num % 2 != 0);

cout << "You entered: "
        << num << ".\n";
    return 0;
}</pre>
```

CSci 127 (Hunter)

Indefinite Loops: do-while

```
General format:

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

Today's Topics

```
//Another C++ program, demonstrating L/O & arithmetic finclude closterous targ namespace std; int main () {
float kg, lbs; 
clin > kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < "lbs * kg * 2.2; 
cut < end < en
```

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

```
//Another C+# program; Demonstrates loops
###Include clostrone std;
int main O
{
   int i, i;
   for (i = 0; i < 4; i+*)
   {
      cout << "The world turned upside down...\n";
   }
   for (j = 18; j > 0; j--)
   {
      cout << y < " ";
      cout << " "i
      cout << " "i
      cout << " "i
      cout << " ";
      cout << " "i
      cout </pre>
```

CSci 127 (Hunter) Lecture 13 12 May 2020 24/39

• I/O: cin >> ...;

```
//Another C++ program; Demonstrates loops
Binclude -dostream
using namespace std;
int min O
{
   int i,j;
   for (i = 0; i < 4; i+>)
   {
      cout < "The world turned upside down...\n";
   }
   for (j = 10; j > 0; j--)
   {
      cout < "s ' = ";
   }
   cout < "8last off!!" << endl;
   return 0;
}</pre>
```

CSci 127 (Hunter) Lecture 13 12 May 2020 24 / 39

0 I/O: cin >> ...; & cout << ...;</pre>

```
//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 12 May 2020 24 / 39

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

```
//Another C+* programs; Demonstrates Loops
###Include clostrong
uning namespace 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 << f > 0; int i,i;
    cout << f > 0; int i,i;
}
```

24 / 39

CSci 127 (Hunter) Lecture 13 12 May 2020

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

```
#Include discharge di
```

//Another C++ program; Demonstrates loops

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

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

```
int main O

[int i,j]

for (3 = 0; i < 4; i++)

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

for (j = 10; j > 0; j--)

{
	cout <= j < c ";
}
	cout <= c "Blast off!!" <= end1;
	return 0;
```

Conditionals:

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

```
\label{eq:controller} / \textit{//Another C+programs; Demonstrates loops} \\ \textit{Enclude -clostrose stdf}, \\ \textit{int main O} \\ \textit{int Lij}, \\ \textit{for (1 = 8; i < 4; i+)} \\ \textit{cout} \ll \text{"The world turned upside down...\n";} \\ \textit{for (3 = 18; j > 8; j--)} \\ \textit{for (3 = 18; j > 8; j--)} \\ \textit{cout} \ll \text{"Bast off!!"} \ll \text{end1;} \\ \textit{return 8;} \\ \end{cases}
```

```
• 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)
                      4 D > 4 A > 4 B > 4 B >
```

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

int main ()

• 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 12 May 2020 26 / 39

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 12 May 2020 26 / 39

Print("Year is", i)

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

Program in C++:

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

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

26 / 39

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) Lecture 13 12 May 2020 26 / 39

• 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>
```

27 / 39

CSci 127 (Hunter) Lecture 13 12 May 2020

• 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)
```

```
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")
```

29 / 39

CSci 127 (Hunter) Lecture 13 12 May 2020

```
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")
```

30 / 39

CSci 127 (Hunter) Lecture 13 12 May 2020

```
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")
```

```
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 \\
     (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")
```

```
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:";
  cin >> 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:";
  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
```

```
#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>
```

```
#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.

• 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()
```

• 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)
```

• 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;
```

• 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;
```

Educational Psychology Study



- If you have consented to participate in the Educational Psychology study, please fill in the 3-question survey
- Clickable link also below the video.
- Thank you for your participation!!!