CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

< ロ ト < 団 ト < 三 ト < 三 ト</p>

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 1 / 34

3

Lecture Slips: tinyurl.com/yc2b6jba



CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

3 29 November 2017 2 / 34

Announcements



• Each lecture includes a survey of computing research and tech in NYC.

Today: Andrew Rasiej CEO/Founder Civic Hall Chair, Hunter CS Advisory Board

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 3 / 34

Announcements



• Each lecture includes a survey of computing research and tech in NYC.

Today: Andrew Rasiej CEO/Founder Civic Hall Chair, Hunter CS Advisory Board

- Today's lecturers include:
 - Prof. Sakas (department chair),
 - Genady Maryash (adjunct coordinator),
 - Katherine Howitt (tutor coordinator).

Today's Topics



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

999

CS Surveys



• Each lecture includes a survey of computing research and tech in NYC.

Today: Andrew Rasiej CEO/Founder Civic Hall Chair, Hunter CS Advisory Board

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 5 / 34

In Pairs or Triples:

Predict what the following pieces of code will do:

```
//Demonstrates conditionals
#include <iostream>
using namespace std;
int main ()
ł
    int vearBorn:
    cout << "Enter year born: ";</pre>
    cin >> yearBorn;
    if (yearBorn < 1946)
         cout << "Greatest Generation":</pre>
    else if (yearBorn <= 1964)
        cout << "Baby Boomer";</pre>
    else if (yearBorn <= 1984)
         cout << "Generation X";</pre>
    else if (vearBorn \leq 2004)
    {
        cout << "Millennial":</pre>
    3
    else
    {
        cout << "TBD";</pre>
    return ∅;
```

```
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 ":
    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>
```

イロト イポト イヨト イヨト

Lecture 13: tinyurl.com/yc2b6jba

= nar

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;
int main ()
£
  float kg, lbs;
  cout << "Enter kg: ";</pre>
  cin >> kg;
  lbs = kg * 2.2;
  cout << endl << "Lbs: " << lbs << "\n\n":
  return 0;
3
```

CSci 127 (Hunter)

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~

• Efficient for systems programming.

```
//Another C++ program, demonstrating I/O & arithmetic
#include cisotream-
using namespace std;
int main O
{ float kg, lbs;
cout << "Enter kg: ";
lbs = kg * 2.2;
cout << endl << "lbs: " << lbs << "\n\n";
return 0;
}
```

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 8 / 34

SOC

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

```
//Another C++ program, demonstrating I/O & arithmetic
wining ammespace std;
int main O
{
    float kg, lbs;
    coat << "Enter kg: ";
    tbs kg * 2.2;
    coat << "enter kg: " << lbs << "\n\n";
    return 0;
```

Sac

- 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 ka, lbs:
 cout << "Enter kg: ";
 cin >> kg;
 lbs = kg * 2.2;
 cout << endl << "Lbs: " << lbs << "\n\n":
 return 0:
3
```

Sac

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

```
//Another C++ program. demonstrating I/O & arithmetic
minclude cisteream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "Enter kg: ";
  cin > kg; 2.2;
  coat << endl << "Lbs: " << lbs << "\n\n";
  return 0;
```

3

Sac

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

```
//Another C++ program. demonstrating I/O & arithmetic
minclude cisteream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "Enter kg: ";
  cin > kg; 2.2;
  coat << endl << "Lbs: " << lbs << "\n\n";
  return 0;
```

3

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;
int main ()
 float ka, lbs:
 cout << "Enter kg: ";
 cin >> kg;
 lbs = kg * 2.2;
 cout << endl << "Lbs: " << lbs << "\n\n":
 return 0:
3
```

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

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

Sac

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

Efficient for systems programming.

- Programs are organized in functions.
- Must declare variables: int num:
- Many types available:

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

To print:

3

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
wining ammespace std;
int main ()
{ float kg, lbs;
coat >< driver kg; ";
coat >< kg; ?;
lbs = kg ? 2.2;
coat << mg; * lbs << "\n\n";</pre>
```

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!!";

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

э

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;
int main ()
 float ka, lbs:
 cout << "Enter ka: ";
 cin >> kg;
 lbs = kg * 2.2;
 cout << endl << "Lbs: " << lbs << "\n\n":
 return 0:
3
```

- 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:

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cins - kg; 2.2;
  coat << endl << "lbs: " << lbs << "\n\n";
  returm 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;

э

Sar

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cins - kg; 2.2;
  coat << endl << "lbs: " << lbs << "\n\n";
  returm 0;
```

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

int, float, char, ...

- To print: cout << "Hello!!";
- To get input: cin >> num;
- To use those I/O functions:

```
//Another C++ program, demonstrating I/O & arithmetic
minclude cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cins - kg; 2.2;
  coat << endl << "lbs: " << lbs << "\n\n";
  returm 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 C++ program, demonstrating I/O & arithmetic
minclude cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cins - kg; 2.2;
  coat << endl << "lbs: " << lbs << "\n\n";
  returm 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:

Lecture 13: tinyurl.com/yc2b6jba

```
//Another C++ program, demonstrating I/O & arithmetic
%include ciostreams
using namespace std;
int main O
{
foot kg, lbs;
cout << "Enter kg: ";
cin >> kg' 2.2;
lbs - kg * 2.2;
return @ eff( <* lbs * < "\n\n";
return % of *;
```

- 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>

Lecture 13: tinyurl.com/yc2b6jba

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
%include ciostreams
using namespace std;
int main O
{
foot kg, lbs;
cout << "Enter kg: ";
cin >> kg' 2.2;
lbs - kg * 2.2;
return @ eff( <* lbs * < "\n\n";
return % of *;
```

- 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++) $\{\ldots\}$

イロト イポト イヨト イヨト 二日

• Blocks of code uses '{' and '}'.

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 8 / 34

Sac

```
//Another C++ program, demonstrating I/O & arithmetic
%include ciostreams
using namespace std;
int main O
{
foot kg, lbs;
cout << "Enter kg: ";
cin >> kg' 2.2;
lbs - kg * 2.2;
return @ eff( <* lbs * < "\n\n";
return % of *;
```

- 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++) $\{\ldots\}$

イロト イポト イヨト イヨト 二日

- Blocks of code uses '{' and '}'.
- Commands generally end in ';'.

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 8 / 34

Sac

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



gdb.org

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 9 / 34

3

900

- 4 ⊒ →

・ロト ・日ト ・ヨト



gdb.org

- Part of Richard Stallman's "GNU is Not Unix" (GNU) project.
- Written in 1986, gdb is the GNU debugger and based on dbx from the Berkeley Distribution of Unix.

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 9 / 34



gdb.org

- Part of Richard Stallman's "GNU is Not Unix" (GNU) project.
- Written in 1986, gdb is the GNU debugger and based on dbx from the Berkeley Distribution of Unix.
- Lightweight, widely-available program that allows you to "step through" your code line-by-line.



gdb.org

- Part of Richard Stallman's "GNU is Not Unix" (GNU) project.
- Written in 1986, gdb is the GNU debugger and based on dbx from the Berkeley Distribution of Unix.
- Lightweight, widely-available program that allows you to "step through" your code line-by-line.
- Available on the lab machines (via command-line and the IDE spyder) and on-line (onlinegdb.com).

$\mathsf{C}{++} \mathsf{Demo}$

```
//Demonstrates conditionals
#include <iostream>
using namespace std:
int main ()
£
    int yearBorn:
    cout << "Enter year born: ";</pre>
    cin >> yearBorn;
    if (yearBorn < 1946)
        cout << "Greatest Generation";</pre>
    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 ∅;
}
```

29 November 2017 10 / 34

3

イロト イポト イヨト イヨト

996

CSci 127 (Hunter)

Conditionals

General format:



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

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 11 / 34

3

Sac

Logical Operators in C++

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

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 12 / 34

<ロト < 同ト < 臣ト < 臣ト = 臣 = のへで

Logical Operators in C++ $\,$

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: tinyurl.com/yc2b6jba

29 November 2017 12 / 34

<ロト < 部 ト < 注 ト < 注 ト 三 三 の < ()</p>

Logical Operators in C++

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	False	False
False	True	True
True	False	True
True	True	True

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 12 / 34

(日) (同) (三) (三) (三) (○) (○)

Logical Operators in C++

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	False	False
False	True	True
True	False	True
True	True	True

not (!)

	in1	returns:
!	False	True
!	True	False

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 12 / 34

(日) (同) (三) (三) (三) (○) (○)

In Pairs or Triples:

Predict what the following pieces of code will do:

```
//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: ";</pre>
      cin >> num;
  3
  cout << "You entered: "
       << num << ".\n";
  return 0;
```

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

```
int main ()
{
    int population = 100;
    int year = 0;
    cout << "Year\tPopulation\n";
    while (population < 1000)
    {
        cout << year << "\t" << population << "\n";
        population = population * 2;
    }
    return 0;
}
    CSci 127 (Hunter) Lecture 13: tinyurl.com/yc2b6jba</pre>
```

```
//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>
```

イロト イポト イヨト イヨト

29 November 2017

= nar

13 / 34

$\mathsf{C}{++} \mathsf{Demo}$

```
//Demonstrates loops
#include <iostream>
using namespace std;
int main ()
  int num;
  cout << "Enter an even number: ";</pre>
  cin >> num;
  while (num \% 2 != 0)
                                               (Demo with onlinegdb)
  {
      cout << "\nThat's odd!\n":</pre>
      cout << "Enter an even number: ";</pre>
      cin >> num;
  3
  cout << "You entered: "
      << num << ".\n";
  return ∅;
```

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

= nar

Indefinite Loops: while

```
//Demonstrates loops
#include <iostreams
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;

. . .

Lecture 13: tinyurl.com/yc2b6jba

= nar

Indefinite Loops: do-while

```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std:
int main ()
  int i.j.size:
  cout << "Enter size: ";</pre>
  cin >> size:
  for (i = 0; i < size; i++)
    for (j = 0; j < size; j++)</pre>
    cout << "*":
    cout << endl:
  cout << "\n\n":
  for (i = size; i > 0; i--)
    for (j = 0; j < i; j++)
    cout << "*":
    cout << endl:
  return 0:
```

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

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 16 / 34

3

Sac

#Name: your name here
#Date: October 2017
#This program, uses functions,
says hello to the world!

def main():
 print("Hello, World!")

if __name__ == "__main__":
 main()

Plan: Alternate between working in pairs and sketching solutions (until time runs out):

```
//Another C++ program: Demonstrates loops
#include <iostream>
using namespace std:
int main ()
  int i.i.size:
  cout << "Enter size: ";
  cin >> size:
  for (i = 0; i < size; i ++)
    for (i = 0; i < size; i++)
    cout << "*":
    cout << endl:
  3
  cout << "\n\n":
  for (i = size; i > 0; i--)
    for (j = 0; j < i; j++)
    cout << "*";
    cout << endl;</pre>
 }
  return 0;
```

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 17 / 34

3

Sac

#Name: your name here
#Date: October 2017
#This program, uses functions,
says hello to the world!

```
def main():
    print("Hello, World!")
```

if __name__ == "__main__": main()

```
//Another C++ program: Demonstrates loops
#include <iostream>
using namespace std:
int main ()
  int i.i.size:
  cout << "Enter size: ";
  cin >> size:
  for (i = 0; i < size; i ++)
    for (i = 0; i < size; i++)
    cout << "*":
    cout << endl:
  3
  cout << "\n\n":
  for (i = size; i > 0; i--)
    for (j = 0; j < i; j++)
    cout << "*";
    cout << endl;</pre>
  3
  return 0;
```

Plan: Alternate between working in pairs and sketching solutions (until time runs out):

• Definite Loops in Python & C++

Lecture 13: tinyurl.com/yc2b6jba

3

Sar

#Name: your name here
#Date: October 2017
#This program, uses functions,
says hello to the world!

```
def main():
    print("Hello, World!")
```

if __name__ == "__main__": main()

```
//Another C++ program: Demonstrates loops
#include <iostream>
using namespace std:
int main ()
  int i.i.size:
  cout << "Enter size: ";
  cin >> size:
  for (i = 0; i < size; i ++)
    for (i = 0; i < size; i++)
     cout << "*":
    cout << endl:
  cout << "\n\n":
  for (i = size; i > 0; i--)
    for (j = 0; j < i; j++)
    cout << "*";
    cout << endl;</pre>
  3
  return 0;
```

Plan: Alternate between working in pairs and sketching solutions (until time runs out):

- Definite Loops in Python & C++
- Conditionals in Python & C++

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 17 / 34

3

#Name: your name here
#Date: October 2017
#This program, uses functions,
says hello to the world!

```
def main():
    print("Hello, World!")
```

if __name__ == "__main__": main()

```
//Another C++ program: Demonstrates loops
#include <iostream>
using namespace std:
int main ()
  int i.i.size:
  cout << "Enter size: ";
  cin >> size:
  for (i = 0; i < size; i ++)
    for (i = 0; i < size; i++)
     cout << "*":
    cout << endl:
  3
  cout << "\n\n";
  for (i = size; i > 0; i--)
    for (j = 0; j < i; j++)
     cout << "*";
    cout << endl;</pre>
  3
  return 0;
```

Plan: Alternate between working in pairs and sketching solutions (until time runs out):

- Definite Loops in Python & C++
- Conditionals in Python & C++
- Indefinite Loops in Python & C++

Lecture 13: tinyurl.com/yc2b6jba

Complete as many as possible:

• Python: what is the output? for i in range(2017, 2000, -2): print("Year is", i)

• In Python, write a complete program that prints out 1 to 100.

• In C++, write a complete program that prints out 1 to 100.

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > <

• Python: what is the output? for i in range(2017, 2000, -2): print("Year is", i)

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

• In Python, write a complete program that prints out 1 to 100.

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 21 / 34

• In C++, write a complete program that prints out 1 to 100.

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 22 / 34

- Write Python code that asks for the hour (24-hour time) and greets then with "Good Morning" if before 12, "Good Afternoon" for 12 but not yet 17, and "Good Evening" otherwise.
- 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 \$31.00, and the cost of single ride, with bonus, is \$2.48).

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!!")
```

<□> <同> <同> < 回> < 回> < 回> < 回> < 回> < 0 < 0

• Write Python code that asks for the hour (24-hour time) and greets then with "Good Morning" if before 12, "Good Afternoon" for 12 but not yet 17, and "Good Evening" otherwise.

Sac

 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 \$31.00, and the cost of single ride, with bonus, is \$2.48).

Sac

In Pairs or Triples: Indefinite Loops in Python & C++ Complete as many as possible:

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 28 / 34

イロト 不良 トイヨト イヨト ヨー のくや

In Pairs or Triples: Indefinite Loops in Python & C++ Complete as many as possible:

```
• Python: what is the output?
  bal = 100
  while bal < 200:
       print("Balance", bal)
       bal = bal + 0.1*bal
• C++: what is the output?
  int n = 10;
  do {
       if (n \% 2 == 0)
          n = n / 2:
       else
          n = 3*n + 1;
       cout << "n is " << endl;</pre>
  } while (n > 1);
```

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

```
• Python: what is the output?
bal = 100
while bal < 200:
    print("Balance", bal)
    bal = bal + 0.1*bal
```

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

```
• C++: what is the output?
int n = 10;
do {
    if ( n % 2 == 0)
        n = n / 2;
    else
        n = 3*n + 1;
    cout << "n is " << endl;
} while (n > 1);
```

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

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

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 31 / 34

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

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 32 / 34

• I/O:

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 33 / 34

Sac

• I/O: cin >> ...;

```
//Arother (-+ program, Benostrates loops 

#iclude -(otreteme)

with memopole tid;

in teal O

( int.jd;

for (1 = 0; i < 4; i+-)

f out << "The world turned upside down...\n";

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

f out << "Elsat off!!" << endl;

return 0;

return 0;
```

Lecture 13: tinyurl.com/yc2b6jba

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 - のの⊙

• I/O: cin >> ...; & cout << ...;

Lecture 13: tinyurl.com/yc2b6jba

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 - のの⊙

• I/O: cin >> ...; & cout << ...;

• Definite loops:

E Sac

• I/O: cin >> ...; & cout << ...;

• Definite loops:



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

```
 \begin{array}{l} (\text{in that } O ) \\ \{ & \text{int } i, j \} \\ f \ (1 - 0; \ i < 4; \ i + ) \\ f \ (\text{cost} < < \text{``the world turned upside down...\vi';} \\ f \ or \ (j = 10; \ j > 0; \ j - .) \\ \ (\text{out } < j < < \text{``the solution } j < 0; \ j < 0; \ j < 0; \ i <
```

= nar

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

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

```
 \begin{array}{l} (\text{in tank } Q) \\ \left\{ \begin{array}{l} \text{int } \{i,j\} \\ \text{for } \{i = 0; \ i < 4; \ i + \ast \} \\ f \end{array} \right. \\ \left\{ \begin{array}{l} \text{cout } \ll \text{ The world turned upside down...\n";} \\ \\ f \end{array} \right. \\ \left\{ \begin{array}{l} \text{for } (j = 10; \ j > 0; \ j - \ast ) \\ \\ \text{cout } \ll \text{ Tank off} \left\{ \begin{array}{l} \text{mod} \\ \text{mod} \\ \text{mod} \\ \text{cout } \ll \text{ Tank off} \right\} \\ \\ \text{cout } \ll \text{ Tank off} \left\{ \begin{array}{l} \end{array} \right. \\ \\ \text{cout } \ll \text{ mod} \\ \text{cout } \\ \end{array} \right. \\ \end{array} \right.
```

= nar

```
//Another (-+ program, Demonstrates loops

wintig memory and the program demonstrates loops

wintig memory and the program demonstrates loops

int noin ()

{ int i, j;

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

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

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

{ cout << "Blast off!!" << endl;

return 0;

}
```

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

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 33 / 34

= nar

```
\label{eq:linear_constraints} $$ cops $$ include dots reconstraints $$ cops $$ include dots reconstraints $$ cops $$ include dots reconstraints $$ include dots reconstraints $$ int nois 0$$ $$ int 1; $$ or (1 = 0; i < 4; i++) $$ for (1 = 0; i < 4; i++) $$ for (1 = 0; i < 0; i--) $$ for (1 = 10; i > 0; j--) $$ for (1 = 10; i > 0; i--) $$ cout < $$ is stored of $$ is off! "< endl; $$ return 0; $$ }$ $$ }
```

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

CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 33 / 34

= nar

```
• I/O: cin >> ...; & cout << ...;
</pre>
                                            Definite loops:
                                               for (i = 0; i < 10; i++)
                                                       ...
                                            Conditionals:
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;
                                               if (logical expression)
int main ()
int i,j;
 for (i = 0; i < 4; i++)
   cout << "The world turned upside down...\n";</pre>
 for (j = 10; j > 0; j - -)
   cout << j << " ":
                                               else
 cout << "Blast off!!" << endl:
 return 0;
                                            Indefinite loops:
                                               while (logical expression)
                                                       ...
                                                                             イロト イポト イヨト イヨト
                                                                                                                   Sac
                                                                                                               3
    CSci 127 (Hunter)
                                        Lecture 13: tinyurl.com/yc2b6jba
                                                                                           29 November 2017
                                                                                                                   33 / 34
```

Lecture Slips: tinyurl.com/yc2b6jba



CSci 127 (Hunter)

Lecture 13: tinyurl.com/yc2b6jba

29 November 2017 34 / 34

3

Sac