CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

• Please take a moment to fill out the Teacher Evaluations

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

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

- Please take a moment to fill out the Teacher Evaluations
- Your chance to provide feedback!
- Responses are completely anonymous.
- Smartphone: www.hunter.cuny.edu/mobilete

- Please take a moment to fill out the Teacher Evaluations
- Your chance to provide feedback!
- Responses are completely anonymous.
- Smartphone: www.hunter.cuny.edu/mobilete
- Computer: www.hunter.cuny.edu/te

- Please take a moment to fill out the Teacher Evaluations
- Your chance to provide feedback!
- Responses are completely anonymous.
- Smartphone: www.hunter.cuny.edu/mobilete
- Computer: www.hunter.cuny.edu/te

• Final Exam December 19 at 9-11 AM

- Final Exam December 19 at 9-11 AM
- Seating assignment available on Blackboard/My Grades
 Please check now, if there are any problems, email me ASAP

- Final Exam December 19 at 9-11 AM
- Seating assignment available on Blackboard/My Grades
 Please check now, if there are any problems, email me ASAP
- Next Tuesday December 13, we will have a Mock Exam

- Final Exam December 19 at 9-11 AM
- Seating assignment available on Blackboard/My Grades
 Please check now, if there are any problems, email me ASAP
- Next Tuesday December 13, we will have a Mock Exam
 - Room 118 Hunter North (Assembly Hall), ground floor of the North Building

- Final Exam December 19 at 9-11 AM
- Seating assignment available on Blackboard/My Grades
 Please check now, if there are any problems, email me ASAP
- Next Tuesday December 13, we will have a Mock Exam
 - Room 118 Hunter North (Assembly Hall), ground floor of the North Building
 - ▶ Only 1.15 hours for the Mock, 2 hours for the real exam.

- Final Exam December 19 at 9-11 AM
- Seating assignment available on Blackboard/My Grades
 Please check now, if there are any problems, email me ASAP
- Next Tuesday December 13, we will have a Mock Exam
 - Room 118 Hunter North (Assembly Hall), ground floor of the North Building
 - ▶ Only 1.15 hours for the Mock, 2 hours for the real exam.
 - Just a practice run, this WILL NOT be the same as the real exam, and it will not be graded.

CSci 127: Introduction to Computer Science

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

What's the best way to study for the final exam?
 The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.

• What's the best way to study for the final exam? The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.

Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through , grade yourself, update note sheet, and repeat.

- What's the best way to study for the final exam? The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
 - Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through , grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit?

- What's the best way to study for the final exam?
 The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
 - Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through , grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study.

- What's the best way to study for the final exam? The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
 - Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through , grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit?
 It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
 - Deadline: Monday, December 5, 2022, 11:59 PM (EST)

- What's the best way to study for the final exam?
 The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
 - Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through , grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
- Deadline: Monday, December 5, 2022, 11:59 PM (EST)

 Why do you care about cheating?

- What's the best way to study for the final exam?
 The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
 - Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through , grade yourself, update note sheet, and repeat.
- I'm worried about my grade. Should I do Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study.
 - Deadline: Monday, December 5, 2022, 11:59 PM (EST)
- Why do you care about cheating?
 First: it gives unfair advantage & is immoral.

What's the best way to study for the final exam?
 The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.

Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through , grade yourself, update note sheet, and repeat.

I'm worried about my grade. Should I do Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study.

Deadline: Monday, December 5, 2022, 11:59 PM (EST)

• Why do you care about cheating?
First: it gives unfair advantage & is im

First: it gives unfair advantage & is immoral.

Second: it degrades the quality of our students.

What's the best way to study for the final exam?
 The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.

Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through , grade yourself, update note sheet, and repeat.

I'm worried about my grade. Should I do Pass/NoCredit? It's fine with us, but check with your advisor to make sure it's accepted for your program of study.

Deadline: Monday, December 5, 2022, 11:59 PM (EST)

• Why do you care about cheating?

First: it gives unfair advantage & is immoral.

Second: it degrades the quality of our students.

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

What's the best way to study for the final exam?
 The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.

Past exams (and answer keys) are on-line. Do 7-10 previous exams: allow 1 hour (half time) and work through , grade yourself, update note sheet, and repeat.

I'm worried about my grade. Should I do Pass/NoCredit?
 It's fine with us, but check with your advisor to make sure it's accepted for your program of study.

Deadline: Monday, December 5, 2022, 11:59 PM (EST)

• Why do you care about cheating?

First: it gives unfair advantage & is immoral.

Second: it degrades the quality of our students.

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

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

5 / 43

CSci 127 (Hunter) Lecture 13 December 6, 2022

Today's Topics

```
//Amother (-» program, demonstrating I/O & arithmetic Britished constraint of the Imminused Constraint of Immi
```

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

Today's Topics

```
//Acother C+p program, demonstrating I/O & arithmetic finitude closterose tst; int main () { | filont kg, lbs; cost < "Enter kg, lbs; cost < "Enter kg,"; lbs < "kg," 2.2; cost < mail < "lbs < "kg," 2.2; cost < mail < "lbs; "kg," 18s; " < lbs < "\n\n"; return 0; return 0;
```

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

CSci 127 (Hunter) Lecture 13

```
Recap: Basic Form & I/O in C++
1 //C++ program demonstrating I/O & arithmetic
2 | #include <iostream>
 using namespace std;
4
  int main ()
    float kg, lbs;
    cout << "Enter kg: ";</pre>
    cin >> kg;
    lbs = kg * 2.2;
10
    cout << endl << "Lbs: " << lbs << "\n\n":
11
    return 0;
12
```

Lecture 13

December 6, 2022

CSci 127 (Hunter)

Efficient for systems programming.

```
//Another C++ program, demostrating I/O & arithmetic simclude ciostream-using namespace std; int main O { floot kg, lbs; cott <= "Enter kg: "; cin >> kg; lbs = kg * 2.2; cott <= end! <= "Lbs: " << "Un\n"; return 0; } }
```

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

```
//Another C++ program, demonstrating I/O & arithmetic finclude cisotreams using namespace std; int main O { floot kg, lbs; cott < "Enter kg: "; cott < "Enter kg: "; cott <> kg: " cott << end! < "Lbs: " << lbs << "\n\n"; return 0; } kg: cott << end! << "Lbs: " << lbs << "\n\n"; return 0; }
```

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

```
//Another C++ program, demonstrating I/O & arithmetic finitude clostreams using nomespace std; int main () { float (kg. lbs; cout << "Enter kg: "; cin >> kg; lbs = kg * 2.2; cout << endl << "Lbs: " << "kh\n"; return 0; }
```

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

```
//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:
```

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

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

CSci 127 (Hunter) Lecture 13

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

```
//Another C++ program, demonstrating I/O & arithmetic
finclude cistremb
using namespace std;
int main C)
{
   float kg, lbs;
   cout << "Enter kg: ";
   cin kg;
   cin kg;
   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:

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

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

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

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

```
//Another (++ program, demonstrating I/O & arithmetic sinclude <lastrambusing namespace std; 
int main O {
    float kg, lbs; 
    cout << "Enter kg: "; 
    cin >> kg; 
    loat 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: int, float, char, ...
- To print: cout << "Hello!!";
- To get input: cin >> num;

```
//Another (++ program, demonstrating I/O & arithmetic sinclude <lastrambusing namespace std; 
int main () {
    float kg, lbs; 
    cout << "Enter kg: "; 
    chassing 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:

```
• 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 finclude -tostream using nonespoce std; int main () { float kg, lbs: cout << "Enter kg: "; cin >> kg; lbs = kg * 2.2; cout << endl << "Lbs: " << lbs << "\n\n"; return 0; }
```

```
    Efficient for systems programming.
```

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

```
//Another C++ program, demonstrating I/O & arithmetic finclude <id>demonstrating I/O & arithmetic finclude 
int main O

{
float kg, lbs;
cout << "Enter kg: ";
Lbs | kg = 2.2;
cout << endl << "Lbs: " << lbs << "\n\n";
return 0;
}
```

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

```
//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:
```

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

```
    Efficient for systems programming.
```

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

```
//Acother (** program, demonstrating L/O & arithmetic finctude closterous transpose Std): international consequence Std; international consequence of flood to go that consequence consequ
```

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

CSci 127 (Hunter)

Challenge:

Predict what the following pieces of code will do:

```
//Demonstrates conditionals
#include <iostream>
using namespace std:
int main ()
    int yearBorn;
    cout << "Enter year born: ";
    cin >> yearBorn;
    if (yearBorn < 1946)
        cout << "Greatest Generation";</pre>
    else if (yearBorn <= 1964)
        cout << "Baby Boomer":
    else if (yearBorn <= 1984)
        cout << "Generation X";</pre>
    else if (vearBorn <= 2004)
        cout << "Millennial":</pre>
    else
        cout << "TBD":
    return 0:
   CSci 127 (Hunter)
```

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

December 6, 2022

11 / 43

Lecture 13

```
//C++ program demonstrates conditionals
     #include <iostream>
     using namespace std;
     int main ()
6
         int yearBorn;
         cout << "Enter year born: ";
         cin >> yearBorn;
10
         if (yearBorn < 1946)
11
             cout << "Greatest Generation":</pre>
13
14
         else if (yearBorn <= 1964)
16
             cout << "Baby Boomer":
17
         else if (vearBorn <= 1984)
19
20
             cout << "Generation X";</pre>
21
         else if (vearBorn <= 2004)
23
             cout << "Millennial";</pre>
         else
27
28
             cout << "TBD":
30
31
         return 0;
32
     }\\
```

Conditionals

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

General format:

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

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

CSci 127 (Hunter) Lecture 13 December 6, 2022 14 / 43

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 December 6, 2022 14 / 43

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

and (&&)

in1		in2	returns:
False	&&	False	False
False	&&	True	False
True	&&	False	False
True	&&	True	True

or (||)

in1		in2	returns:
False		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
or ()			

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

not (!)

in1		returns:	
<u>!</u>	False	True	
!	True	False	

Lecture Slip

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

CSci 127 (Hunter) Lecture 13 December 6, 2022 15 / 43

Today's Topics

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

Challenge:

Predict what the code will do: https://www.onlinegdb.com/rk86urUlf

```
//While Growth example
   #include <iostream>
   using namespace std;
4
   int main ()
5
6
     int population = 100;
7
     int year = 0;
8
     cout << "Year\tPopulation\n";</pre>
9
     while (population < 1000)</pre>
10
11
          cout << year << "\t" << population << "\n";</pre>
12
          population = population * 2;
13
14
     return 0;
15
16
     CSci 127 (Hunter)
                                 Lecture 13
                                                       December 6, 2022
                                                                    17 / 43
```

C++ Demo

```
///Mhile Growth Example
#include <iostream>
using namespace std;

int main ()
{
    int population = 100;
    int year = 0;
    cout << "Year\tPopulation\n";
    white(population < 1000)
    {
        cout << year << "\t\t\t" << population = population * 2;
        year++;
    }
    return 0;
}</pre>
```

link: https://www.onlinegdb.com/rk86urUlf

Indefinite Loops: while

```
///White 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\t" << population << "\n";
    population = population * 2;
   year++;
   }
   return 0;
}</pre>
```

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

Challenge: predict what the code do

```
#include <iostream>
   using namespace std;
3
   int main ()
      int num;
     cout << "Enter an even number: ";</pre>
     cin >> num;
     while (num % 2 != 0)
     {
10
          cout << "\nThat's odd!\n";</pre>
11
          cout << "Enter an even number: ";</pre>
12
          cin >> num;
13
14
     cout << "You entered: " << num << ".\n";</pre>
15
     return 0;
16
17
      CSci 127 (Hunter)
                                   Lecture 13
                                                           December 6, 2022
                                                                         20 / 43
```

C++ Demo

https://www.onlinegdb.com/rJttLSLgG

CSci 127 (Hunter)

Indefinite Loops: while

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

```
Challenge: predict what the code will do
   //Demonstrates do-while loops
   #include <iostream>
   using namespace std;
4
   int main ()
6
     int num;
7
     do
         cout << "Enter an even number: ";</pre>
10
         cin >> num:
11
     } while (num % 2 != 0);
12
13
     cout << "You entered: " << num << ".\n";</pre>
14
     return 0;
15
16
```

Lecture 13

December 6, 2022

23 / 43

CSci 127 (Hunter)

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>
```

link: https://www.onlinegdb.com/Bkn8DB8eG

CSci 127 (Hunter) Lecture 13

24 / 43

Indefinite Loops: do-while

```
General format:

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

Today's Topics

```
//Acother (** program, demonstrating L/O & arithmetic film(used conternal members tid) intended to the contended of the conte
```

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

CSci 127 (Hunter)

${\sf Recap:}\ C++\ {\sf Control\ Structures}$

I/O:

```
//Arother C+s program; Demonstrates loops finclude clostered include clostered inclu
```

CSci 127 (Hunter) Lecture 13 December 6, 2022 27 / 43

I/O: cin >> ...;

```
//Another C++ program; formostrates loops functioned controverses std; fun
```

CSci 127 (Hunter) Lecture 13 December 6, 2022 27 / 43

• I/O: cin >> ...; & cout << ...;

```
//Another C+s program; Demonstrates loops finclude clostyre commission gramspose std; ten and () {
    (st i, j; for i < 4; i++) {
        cout < "The world turned upside down...\v"; }
    for (j = 18; j > 8; j--) {
        cout < 'g < ""; }
    cout < "Blast off!!" << endl; return 8; }
```

CSci 127 (Hunter) Lecture 13 December 6, 2022 27 / 43

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

```
//Another C+program; Demonstrates loops entroduce dostrone such grammapace atd; int main O { int i,i; for (i \to 0, i \to 4, i \mapsto) { cost << "The world turned upside down...\n"; } for (j \to 0, i \to 4, i \mapsto) { cost << "The world turned upside down...\n"; } for (j \to 0, i \to 4, i \mapsto) { cost << j <= "; } cost << "Blast off!!" << end1; return 0, i \to 0; retu
```

27 / 43

CSci 127 (Hunter) Lecture 13 December 6, 2022

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

```
//Another C+r program; Demonstrates loops principle dostrooms with grammapace sid; jith enin ()

int i,j; for (i = 0; i < 4; i+·) {
    cout << "The world turned upside down...\n"; }

for (j = 10; j > 0; j-·) {
    cout << "Slast off!" << end!; return 0; }

cout << "Slast off!" << end!; return 0;
```

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

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

```
int main ()  \begin{cases} &\text{int } i, j; \\ &\text{for } (i = 0; i < 4; i+\epsilon) \\ &\text{cout} << \text{'The world turned upside down...\n'';} \\ &\text{for } (j = 10; j > 0; j--) \\ &\text{cout} << j << - ''; \\ &\text{cout} << c ** instance of the cond; \\ &\text{return } 0; \end{aligned}
```

Conditionals:

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

```
//Monther C++ programs, Demonstrates loops
include clostream
using nomespace std;
int main ()

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

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

Indefinite loops:

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

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

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

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

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

Recap: C++ Control Structures

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

using nonespace std; int moin O (int $\{j,j\}$) for $(i=0;\ i<1;\ i+1\}$) for $(i=0;\ i<1;\ i+1)$ (out $i=1,\dots,n$) for $(j=10;\ j>0;\ j=0;\ j=0)$

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

cout << j << " ":

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

• 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>
```

December 6, 2022

• Rewrite this program in C++:

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

CSci 127 (Hunter) Lecture 13 December 6, 2022 29 / 43

Print("Year is", i)

#include <iostream>
using namespace std;

 Rewrite this program in C++:

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

#include <iostream>
using namespace std;

CSci 127 (Hunter) Lecture 13 December 6, 2022 29 / 43

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()

29 / 43

Print("Year is", i)

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

for (int i = 2017; i > 2000; i=i-2)

29 / 43

• 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;
```

29 / 43

• Rewrite this program in Python:

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

CSci 127 (Hunter) Lecture 13 December 6, 2022 30 / 43

• Rewrite this program in Python:

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

30 / 43

CSci 127 (Hunter) Lecture 13 December 6, 2022

• 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")
```

CSci 127 (Hunter) Lecture 13 December 6, 2022 32 / 43

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

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

CSci 127 (Hunter) Lecture 13 December 6, 2022 33 / 43

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

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

CSci 127 (Hunter) Lecture 13 December 6, 2022 34 / 43

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

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

CSci 127 (Hunter) Lecture 13 December 6, 2022 35 / 43

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

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

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

Prints: Leap! Year

38 / 43

• Your program should then print if it is cheaper to buy single ride metro cards (\$2.75 per ride) or 7-day unlimited card (\$33.00).

```
#include <iostream>
using namespace std;
```

CSci 127 (Hunter) Lecture 13 December 6, 2022 39 / 43

• Your program should then print if it is cheaper to buy single ride metro cards (\$2.75 per ride) or 7-day unlimited card (\$33.00).

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

CSci 127 (Hunter) Lecture 13 December 6, 2022 39 / 43

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

```
#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;
```

CSci 127 (Hunter) Lecture 13 December 6, 2022 39 / 43

• Your program should then print if it is cheaper to buy single ride metro cards (\$2.75 per ride) or 7-day unlimited card (\$33.00).

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

CSci 127 (Hunter) Lecture 13 December 6, 2022 39 / 43

```
#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";
```

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";
  return 0;
```

Lecture 13

December 6, 2022

39 / 43

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

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

CSci 127 (Hunter) Lecture 13 December 6, 2022 40 / 43

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

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

```
g = ""
```

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

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

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

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

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

```
s = ""
while s == "":
    s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

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

```
s = ""
while s == "":
    s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

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

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

```
s = ""
while s == "":
    s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

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

```
#include <iostream>
using namespace std;
```

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

```
s = ""
while s == "":
    s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

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

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

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

```
s = ""
while s == "":
    s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

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

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

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

```
s = ""
while s == "":
    s = input("Enter a non-empty string: ")
print("You entered: ", s)
```

• 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)
```

• 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>
```

• 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)
```

• 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.

```
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;
```

December 6, 2022



Before next lecture, don't forget to:

Work on this week's Online Lab



Before next lecture, don't forget to:

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



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001G Hunter North
- Submit this week's 5 programming assignments (programs 55-60)



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001G Hunter North
- Submit this week's 5 programming assignments (programs 55-60)
- If you need help, schedule an appointment for Tutoring in lab 1001G 11:30am-5pm



Before next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001G Hunter North
- Submit this week's 5 programming assignments (programs 55-60)
- If you need help, schedule an appointment for Tutoring in lab 1001G 11:30am-5pm
- Take the Lecture Preview on Blackboard on Monday (or no later than 10:15am on Tuesday)

Lecture Slips & Writing Boards



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