CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

< □ > < □ > < □ > < □ > < □ >

CSci 127 (Hunter)

Lecture 12

July 2021 1 / 78

990

From email and tutoring.

• How can I get info about CS opportunities?

From email and tutoring.

How can I get info about CS opportunities? Join a club! Subscribe to the CUNY2X newsletter.

イロト 不良 トイヨト イヨト ヨー のくや

From email and tutoring.

- How can I get info about CS opportunities? Join a club! Subscribe to the CUNY2X newsletter.
- I want to learn more- what should I take next?

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

From email and tutoring.

- How can I get info about CS opportunities? Join a club! Subscribe to the CUNY2X newsletter.
- I want to learn more- what should I take next?
 - ► Majors: CSci 135 (Software Design and Analysis in C++) & CSci 150 (Discrete Structures)

200

イロト 不得 トイヨト イヨト 二日

From email and tutoring.

- How can I get info about CS opportunities? Join a club! Subscribe to the CUNY2X newsletter.
- I want to learn more- what should I take next?
 - ► Majors: CSci 135 (Software Design and Analysis in C++) & CSci 150 (Discrete Structures)
 - ► Minors: CSci 133 (More Python) & CSci 232 (Databases)

Sac

• 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.

SOC

 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 at least one previous exam: allow 1 hour and work through, grade yourself, update note sheet, and repeat.

- What's the best way to study for the final exam? The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews. Past exams (and answer keys) are on-line. Do at least one previous exam: 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 at least one previous exam: 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 at least one previous exam: 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.

- 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 at least one previous exam: allow 1 hour and work through, grade yourself, update note sheet, and repeat.
- Why do you care about cheating?
 First: it gives unfair advantage & is immoral.
 Second: it degrades the quality of our students.
 Third: it's a standard question on faculty references.

 What's the best way to study for the final exam? The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.
 Past exams (and answer keys) are on-line. Do at least one previous exam: 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.

From our Syllabus.

Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The College is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures. All incidents of cheating will be reported to the Office of Student Conduct in the Vice President for Student Affairs and Dean of Students office.

<ロト < 同ト < 巨ト < 巨ト = 三 の < ○</p>

From our Syllabus.

Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The College is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures. All incidents of cheating will be reported to the Office of Student Conduct in the Vice President for Student Affairs and Dean of Students office.

• All suspected cases of cheating on the final exam (e.g. answer for a different version of the exam) will be reported.

From our Syllabus.

Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The College is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures. All incidents of cheating will be reported to the Office of Student Conduct in the Vice President for Student Affairs and Dean of Students office.

- All suspected cases of cheating on the final exam (e.g. answer for a different version of the exam) will be reported.
- Students will get a PEN grade until the investigation is complete. This may delay registration.

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

From our Syllabus.

Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The College is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures. All incidents of cheating will be reported to the Office of Student Conduct in the Vice President for Student Affairs and Dean of Students office.

- All suspected cases of cheating on the final exam (e.g. answer for a different version of the exam) will be reported.
- Students will get a PEN grade until the investigation is complete. This may delay registration.
- If the student is found in violation by the Office of Student Conduct, they will receive a 0 on the exam, which also means they will fail the class.

CSci 127 (Hunter)

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

Today's Topics



- Recap: Incrementer Design Challenge
- C++: Basic Format & Variables
- I/O and Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python
- More Info on the Final Exam

Sac

Today's Topics



Recap: Incrementer Design Challenge

- C++: Basic Format & Variables
- I/O and Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python
- More Info on the Final Exam

3 July 2021 6 / 78

9990



• Simplest arithmetic: add one ("increment") a variable.



- Simplest arithmetic: add one ("increment") a variable.
- Example: Increment a decimal number:



- Simplest arithmetic: add one ("increment") a variable.
- Example: Increment a decimal number:

```
def addOne(n):
    m = n+1
    return(m)
```



- Simplest arithmetic: add one ("increment") a variable.
- Example: Increment a decimal number:

```
def addOne(n):
    m = n+1
    return(m)
```

• Challenge: Write an algorithm for incrementing numbers expressed as words.



- Simplest arithmetic: add one ("increment") a variable.
- Example: Increment a decimal number:

```
def addOne(n):
    m = n+1
    return(m)
```

 Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"



- Simplest arithmetic: add one ("increment") a variable.
- Example: Increment a decimal number:

```
def addOne(n):
    m = n+1
    return(m)
```

 Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
 Hint: Convert to numbers, increment, and convert back to strings.

CSci 127 (Hunter)

Lecture 12

July 2021 7 / 78



- Simplest arithmetic: add one ("increment") a variable.
- Example: Increment a decimal number:

```
def addOne(n):
    m = n+1
    return(m)
```

- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
 Hint: Convert to numbers, increment, and convert back to strings.
- Challenge: Write an algorithm for incrementing binary numbers.

CSci 127 (Hunter)

Lecture 12

July 2021 7 / 78



- Simplest arithmetic: add one ("increment") a variable.
- Example: Increment a decimal number:

```
def addOne(n):
    m = n+1
    return(m)
```

- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
 Hint: Convert to numbers, increment, and convert back to strings.
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"

CSci 127 (Hunter)

Lecture 12

July 2021 7 / 78



 Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"
- Hint: Convert to numbers, increment, and convert back to strings.



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"
- Hint: Convert to numbers, increment, and convert back to strings.

Pseudocode same for both questions:

Get user input.



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"
- Hint: Convert to numbers, increment, and convert back to strings.

Pseudocode same for both questions:

- Get user input.
- 2 Convert to standard decimal number.



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"
- Hint: Convert to numbers, increment, and convert back to strings.

Pseudocode same for both questions:

- Get user input.
- 2 Convert to standard decimal number.
- 3 Add one (increment) the standard decimal number.



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"
- Hint: Convert to numbers, increment, and convert back to strings.

Pseudocode same for both questions:

- Get user input.
- 2 Convert to standard decimal number.
- 3 Add one (increment) the standard decimal number.
- ④ Convert back to your format.

CSci 127 (Hunter)

Lecture 12

July 2021 8 / 78



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"
- Hint: Convert to numbers, increment, and convert back to strings.

Pseudocode same for both questions:

- Get user input.
- 2 Convert to standard decimal number.
- 3 Add one (increment) the standard decimal number.
- ④ Convert back to your format.
- 9 Print the result.

CSci 127 (Hunter)



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"

Pseudocode same for both questions:

① Get user input: "forty one"



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"

Pseudocode same for both questions:

- Get user input: "forty one"
- ② Convert to standard decimal number: 41

July 2021 9 / 78



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"

Pseudocode same for both questions:

- Get user input: "forty one"
- ② Convert to standard decimal number: 41
- 3 Add one (increment) the standard decimal number: 42



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"

Pseudocode same for both questions:

- Get user input: "forty one"
- ② Convert to standard decimal number: 41
- 3 Add one (increment) the standard decimal number: 42
- ④ Convert back to your format: "forty two"

CSci 127 (Hunter)

Lecture 12

July 2021 9 / 78



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"

Pseudocode same for both questions:

- Get user input: "forty one"
- ② Convert to standard decimal number: 41
- 3 Add one (increment) the standard decimal number: 42
- ④ Convert back to your format: "forty two"
- 9 Print the result.

CSci 127 (Hunter)

Lecture 12

July 2021 9 / 78



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"

Pseudocode same for both questions:

Get user input: "1001"



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"

Pseudocode same for both questions:

- Get user input: "1001"
- ② Convert to standard decimal number: 9

July 2021 10 / 78



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"

Pseudocode same for both questions:

- Get user input: "1001"
- ② Convert to standard decimal number: 9
- 3 Add one (increment) the standard decimal number: 10

CSci 127 (Hunter)

Lecture 12

July 2021 10 / 78



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"

Pseudocode same for both questions:

- Get user input: "1001"
- ② Convert to standard decimal number: 9
- 3 Add one (increment) the standard decimal number: 10
- ④ Convert back to your format: "1010"

CSci 127 (Hunter)

Lecture 12

July 2021 10 / 78



- Challenge: Write an algorithm for incrementing numbers expressed as words. Example: "forty one" → "forty two"
- Challenge: Write an algorithm for incrementing binary numbers. Example: "1001" \rightarrow "1010"

Pseudocode same for both questions:

- Get user input: "1001"
- ② Convert to standard decimal number: 9
- 3 Add one (increment) the standard decimal number: 10
- ④ Convert back to your format: "1010"
- 5 Print the result.

CSci 127 (Hunter)

July 2021 10 / 78



Focus on: Convert to standard decimal number:

Sac



Focus on: Convert to standard decimal number: def convert2Decimal(numString):



Focus on: Convert to standard decimal number: def convert2Decimal(numString): #Start with one-digit numbers: zero,one,...,nine

200



```
Focus on: Convert to standard decimal number:
def convert2Decimal(numString):
    #Start with one-digit numbers: zero,one,...,nine
    if numString == "zero":
        return(0)
```

Sac

イロト 不得下 イヨト イヨト 二日



```
Focus on: Convert to standard decimal number:
def convert2Decimal(numString):
    #Start with one-digit numbers: zero,one,...,nine
    if numString == "zero":
        return(0)
    elif numString == "one":
        return(1)
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○



```
Focus on: Convert to standard decimal number:
def convert2Decimal(numString):
    #Start with one-digit numbers: zero,one,...,nine
    if numString == "zero":
        return(0)
    elif numString == "one":
        return(1)
    elif numString == "two":
        return(2)
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○



```
Focus on: Convert to standard decimal number:
def convert2Decimal(numString):
    #Start with one-digit numbers: zero,one,...,nine
    if numString == "zero":
        return(0)
    elif numString == "one":
        return(1)
    elif numString == "two":
        return(2)
    else:
        return(9)
```

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



```
Focus on: Convert to standard decimal number:
def convert2Decimal(numString):
    #Start with one-digit numbers: zero,one,...,nine
    if numString == "zero":
        return(0)
    elif numString == "one":
        return(1)
    elif numString == "two":
        return(2)
    else:
        return(9)
```

Will this work?

CSci 127 (Hunter)

July 2021 11 / 78

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○



Focus on: Convert to standard decimal number: def convert2Decimal(numString):

```
#Start with one-digit numbers: zero,one,...,nine
if numString == "zero":
    return(0)
elif numString == "one":
    return(1)
elif numString == "two":
    return(2)
else:
    return(9)
```

Will this work? What inputs would find the error(s)?

CSci 127 (Hunter)

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○



Focus on: Convert to standard decimal number: def convert2Decimal(numString):

```
#Start with one-digit numbers: zero,one,...,nine
if numString == "zero":
    return(0)
elif numString == "one":
    return(1)
elif numString == "two":
    return(2)
else:
    return(9)
```

Will this work? What inputs would find the error(s)?

Unit Testing: testing individual units/functions/blocks of code to verify correctness.

CSci 127 (Hunter)

Lecture 12

July 2021 12 / 78



Focus on: Convert to standard decimal number: def convert2Decimal(numString):

```
#Start with one-digit numbers: zero,one,...,nine
if numString == "zero":
    return(0)
elif numString == "one":
    return(1)
elif numString == "two":
    return(2)
else:
    return(9)
```

Will this work? What inputs would find the error(s)?

Unit Testing: testing individual units/functions/blocks of code to verify correctness. Often automated (e.g. gradescope).

CSci 127 (Hunter)

Lecture 12

July 2021 12 / 78



- Unit Testing: testing individual units/functions/blocks of code to verify correctness. Often automated (e.g. gradescope).
- To test all branches of code, would need to test all inputs: "zero", "one",..., "nine", & some bad inputs.



- Unit Testing: testing individual units/functions/blocks of code to verify correctness. Often automated (e.g. gradescope).
- To test all branches of code, would need to test all inputs: "zero", "one",..., "nine", & some bad inputs. Also important to test edge cases.



- Unit Testing: testing individual units/functions/blocks of code to verify correctness. Often automated (e.g. gradescope).
- To test all branches of code, would need to test all inputs: "zero", "one",..., "nine", & some bad inputs. Also important to test edge cases.
- If large, design automated tests that will "cover" as many branches as possible and use randomly generated inputs:



- Unit Testing: testing individual units/functions/blocks of code to verify correctness. Often automated (e.g. gradescope).
- To test all branches of code, would need to test all inputs: "zero", "one",..., "nine", & some bad inputs. Also important to test edge cases.
- If large, design automated tests that will "cover" as many branches as possible and use randomly generated inputs:

names = ["zero","one",...,"nine"]

CSci 127 (Hunter)

July 2021 13 / 78



- Unit Testing: testing individual units/functions/blocks of code to verify correctness. Often automated (e.g. gradescope).
- To test all branches of code, would need to test all inputs: "zero", "one",..., "nine", & some bad inputs. Also important to test edge cases.
- If large, design automated tests that will "cover" as many branches as possible and use randomly generated inputs:

```
names = ["zero","one",...,"nine"]
x = random.randrange(10)
```

CSci 127 (Hunter)

July 2021 13 / 78



- Unit Testing: testing individual units/functions/blocks of code to verify correctness. Often automated (e.g. gradescope).
- To test all branches of code, would need to test all inputs: "zero", "one",..., "nine", & some bad inputs. Also important to test edge cases.
- If large, design automated tests that will "cover" as many branches as possible and use randomly generated inputs:

```
names = ["zero","one",...,"nine"]
x = random.randrange(10)
if x == convert2Decimal(names[x]):
```

CSci 127 (Hunter)

July 2021 13 / 78

イロト 不得下 イヨト イヨト 二日



- Unit Testing: testing individual units/functions/blocks of code to verify correctness. Often automated (e.g. gradescope).
- To test all branches of code, would need to test all inputs: "zero", "one",..., "nine", & some bad inputs. Also important to test edge cases.
- If large, design automated tests that will "cover" as many branches as possible and use randomly generated inputs:

```
names = ["zero","one",...,"nine"]
x = random.randrange(10)
if x == convert2Decimal(names[x]):
    #PASS
```

CSci 127 (Hunter)

July 2021 13 / 78

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙



- Unit Testing: testing individual units/functions/blocks of code to verify correctness. Often automated (e.g. gradescope).
- To test all branches of code, would need to test all inputs: "zero", "one",..., "nine", & some bad inputs. Also important to test edge cases.
- If large, design automated tests that will "cover" as many branches as possible and use randomly generated inputs:

```
names = ["zero","one",...,"nine"]
x = random.randrange(10)
if x == convert2Decimal(names[x]):
    #PASS
else:
```

CSci 127 (Hunter)

July 2021 13 / 78

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



- Unit Testing: testing individual units/functions/blocks of code to verify correctness. Often automated (e.g. gradescope).
- To test all branches of code, would need to test all inputs: "zero", "one",..., "nine", & some bad inputs. Also important to test edge cases.
- If large, design automated tests that will "cover" as many branches as possible and use randomly generated inputs:

```
names = ["zero","one",...,"nine"]
x = random.randrange(10)
if x == convert2Decimal(names[x]):
    #PASS
else:
    #FAIL
```

CSci 127 (Hunter)

July 2021 13 / 78

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

Today's Topics



- Recap: Incrementer Design Challenge
- C++: Basic Format & Variables
- I/O and Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python
- More Info on the Final Exam

900

Challenge:

• Using what you know from Python, predict what the C++ code will do:

```
//Another C++ program, demonstrating variables
 2
   #include <iostream>
   using namespace std;
   int main ()
6 - {
    int year;
   cout << "Enter a number: ";</pre>
     cin >> year;
     cout << "Hello " << year << "!!\n\n";
11
      return 0;
12
```

- 4 伊ト 4 ヨト 4

onlinegdb demo

int main () int year:

(Demo with onlinegdb)

Sac

| | <pre>//Another C++ program, demonstrating variables #include <iostream> using namespace std;</iostream></pre> |
|--|---|
| | int main () { |
| | int year; |
| | <pre>cout << "Enter a number: ";</pre> |
| | cin >> year; |
| | <pre>cout << "Hello << year << "!!\n\n";</pre> |
| | return 0. |

• C++ is a popular programming language that extends C.

200

1 //Mnother C++ program, demonstrating variables
2 #Include -iostreams
3 using nomespace std;
4 int main Q
6 int main Q
6 int space;
7 cout << "Inter a number: ";
8 cout << "Inter a number: ";
9 cin> year;
10 cout << "Hello [! << year << "!!\n\n";
11 return 0;</pre>

- C++ is a popular programming language that extends C.
- Fast, efficient, and powerful.



- C++ is a popular programming language that extends C.
- Fast, efficient, and powerful.
- Used for systems programming (and future courses!).



- C++ is a popular programming language that extends C.
- Fast, efficient, and powerful.
- Used for systems programming (and future courses!).
- Today, we'll introduce the basic structure and simple input/output (I/O) in C/C++.

• Programs are organized in functions.

| 3 | using namespace std; |
|----------|--|
| 4 | |
| 5 | int main () |
| 6- | |
| 6 - 7 | int year; |
| 8 9 | cout << "Enter a number: "; |
| 9 | cin >> year; |
| 10 | <pre>cout << "Hello " << year << "!!\n\n";</pre> |
| 11 12 | |
| 12 | |

996

• Programs are organized in functions.

| using namespace std; |
|--|
| |
| int main () |
| |
| int year; |
| <pre>cout << "Enter a number: ";</pre> |
| cin >> year; |
| <pre>cout << "Hello " << year << "!!\n\n";</pre> |
| |

Example:

999

• Programs are organized in functions.

| using namespace std; |
|--|
| |
| int main () |
| |
| int year; |
| <pre>cout << "Enter a number: ";</pre> |
| cin >> year; |
| cout << "Hello << year << "!!\n\n"; |
| |

Example:

int main()

9990

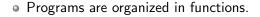
• Programs are organized in functions.

| using namespace std; |
|--|
| |
| int main () |
| |
| int year; |
| <pre>cout << "Enter a number: ";</pre> |
| cin >> year; |
| <pre>cout << "Hello << year << "!!\n\n";</pre> |
| |

Example:

int main() {

9990



| | //Another C++ program, demonstrating variables |
|--------|--|
| 2 | <pre>#include <iostream></iostream></pre> |
| 3 | using namespace std; |
| 4 | |
| 5 | int main () |
| 6- | |
| 7 | int year; |
| 8 9 | <pre>cout << "Enter a number: ";</pre> |
| 9 | cin >> year; |
| 10 | cout << "Hello " << year << "!!\n\n"; |
| 11 | |

Example:

{

}

```
int main()
    cout << "Hello world!";</pre>
    return(0);
```

イロト イポト イヨト イヨト

200

• Programs are organized in functions.

| | using namespace std; |
|--|---------------------------------------|
| | |
| | int main () |
| | |
| | int year; |
| | cout << "Enter a number: ": |
| | cin >> vear: |
| | cout << "Hello " << year << "!!\n\n"; |
| | |
| | |
| | |

999

- Programs are organized in functions.
- Variables must be declared:

| 1 | //Another C++ program, demonstrating variables |
|---|--|
| | |
| | using namespace std; |
| | |
| | int main () |
| | |
| | int year; |
| | cout << "Enter a number: "; |
| | cin >> year; |
| | cout << "Hello " << year << "!!\n\n"; |
| | |
| | |

200

< □ > < □ > < □ > < □ > < □ >

- Programs are organized in functions.
- Variables must be declared:

int num;

| 1 | //Another C++ program, demonstrating variables |
|---|--|
| | |
| | using namespace std; |
| | |
| | int main () |
| | |
| | int year; |
| | cout << "Enter a number: ": |
| | cin >> vear: |
| | cout << "Hello " << year << "!!\n\n"; |
| | return 0: |
| | |

596

| 1 2 3 4 | <pre>//Another C++ program, demonstrating variables #include <iostream> using namespace std;</iostream></pre> |
|------------------|---|
| 5 | int main () |
| 6. | |
| 7 | int year; |
| 8 | cout << "Enter a number: ": |
| 9 | cin >> year; |
| 10 | cout << "Hello " << year << "!!\n\n"; |
| 11 | |
| 12 | |

- Programs are organized in functions.
- Variables must be declared: int num;
- Many types available:
 - int, float, char, ...

Sac

| 2 | <pre>//Another C++ program, demonstrating variables #include <iostream></iostream></pre> |
|--------|--|
| 3 4 | using namespace std; |
| 5 | int main () |
| 6- | |
| 7 | int year; |
| 8 9 | <pre>cout << "Enter a number: ";</pre> |
| | cin >> year; |
| 10 | cout << "Hello " << year << "!!\n\n"; |
| 11 | |
| 12 | |

- Programs are organized in functions.
- Variables must be **declared**: int num;
- Many types available: int, float, char, ...
- Semicolons separate commands:

```
1 //Inother C++ program, demonstrating variables
2 #Include clostneams
3 using numespace std;
4 int main ()
5 int main ()
6 int year;
8 cout << "Enter a number: ";
9 cin > year;
10 cout << "Hello | < year << "!!\n\n";
11 return 0;</pre>
```

- Programs are organized in functions.
- Variables must be **declared**: int num;
- Many types available: int, float, char, ...
- Semicolons separate commands:

num = 5; more = 2*num;

Sar

| | //Another C++ program, demonstrating variables |
|--|--|
| | |
| | using namespace std; |
| | |
| | int main () |
| | |
| | int year; |
| | <pre>cout << "Enter a number: ";</pre> |
| | cin >> year; |
| | <pre>cout << "Hello " << year << "!!\n\n";</pre> |
| | |
| | |

- Programs are organized in functions.
- Variables must be **declared**: int num;
- Many types available: int, float, char, ...
- Semicolons separate commands:

num = 5; more = 2*num;

• To print, we'll use cout <<:

| | using namespace std; |
|--|---------------------------------------|
| | |
| | int main () |
| | |
| | int year; |
| | cout << "Enter a number: ": |
| | cin >> vear: |
| | cout << "Hello " << vear << "!!\n\n": |
| | return 0: |
| | |

- Programs are organized in functions.
- Variables must be **declared**: int num;
- Many types available: int, float, char, ...
- Semicolons separate commands:
 num = 5; more = 2*num;
- To print, we'll use cout <<: cout << "Hello!!";

| | using namespace std; |
|--|--|
| | |
| | int main () |
| | |
| | int year; |
| | <pre>cout << "Enter a number: ";</pre> |
| | cin >> year; |
| | <pre>cout << "Hello " << year << "!!\n\n";</pre> |
| | |
| | |

- Programs are organized in functions.
- Variables must be **declared**: int num;
- Many types available: int, float, char, ...
- Semicolons separate commands: num = 5; more = 2*num;
- To print, we'll use cout <<: cout << "Hello!!";
- To get input, we'll use cin >>:

| | <pre>//Another C++ program, demonstrating variables</pre> |
|--|--|
| | |
| | using namespace std; |
| | |
| | int main () |
| | |
| | int year; |
| | <pre>cout << "Enter a number: ";</pre> |
| | cin >> year; |
| | <pre>cout << "Hello " << year << "!!\n\n";</pre> |
| | |
| | |

- Programs are organized in functions.
- Variables must be **declared**: int num;
- Many types available: int, float, char, ...
- Semicolons separate commands: num = 5; more = 2*num;
- To print, we'll use cout <<: cout << "Hello!!";
- To get input, we'll use cin >>: cin >> num;

| | using namespace std; |
|--|--|
| | |
| | int main () |
| | |
| | int year; |
| | <pre>cout << "Enter a number: ";</pre> |
| | cin >> year; |
| | <pre>cout << "Hello " << year << "!!\n\n";</pre> |
| | |
| | |

- Programs are organized in functions.
- Variables must be **declared**: int num;
- Many types available: int, float, char, ...
- Semicolons separate commands: num = 5; more = 2*num;
- To print, we'll use cout <<: cout << "Hello!!";
- To get input, we'll use cin >>: cin >> num;
- To use those I/O functions, we put at the top of the program:

| | using namespace std; |
|--|--|
| | |
| | int main () |
| | |
| | int year; |
| | cout << "Enter a number: "; |
| | cin >> year; |
| | <pre>cout << "Hello " << year << "!!\n\n";</pre> |
| | |
| | |

- Programs are organized in functions.
- Variables must be **declared**: int num;
- Many types available: int, float, char, ...
- Semicolons separate commands: num = 5; more = 2*num;
- To print, we'll use cout <<: cout << "Hello!!";
- To get input, we'll use cin >>: cin >> num;
- To use those I/O functions, we put at the top of the program: #include <iostream> using namespace std;

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

Challenge:

Predict what the following pieces of code will do:

```
//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:
}
```

July 2021 20 / 78

= nac

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



gdb.org

CSci 127 (Hunter)

Lecture 12

3 July 2021 21 / 78

990

イロト イボト イヨト イヨ



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.

< D > < P > < P > < P >



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-line (onlinegdb.com) or follow installation instructions in Lab 12.

Image: A match a ma

July 2021 21 / 78

C++ Demo

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;
```

```
int moin O
{
    flot kg, lbs;
    cout << "Enter kg: ";
    cin >> kg;
    lbs - kg 2.2;
    cos - dedd << "Lbs: " << lbs << "\n\n";
}</pre>
```

(Demo with onlinegdb)

Challenge:...

Convert the C++ code to a **Python program***:*

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

= nac

Python Tutor

Convert the C++ code to a **Python program***:*

```
//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:
}
```

(Write from scratch in pythonTutor.)

CSci 127 (Hunter)

Lecture 12

July 2021 24 / 78

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~

Today's Topics



- Recap: Incrementer Design Challenge
- C++: Basic Format & Variables
- I/O and Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python
- More Info on the Final Exam

3 July 2021 25 / 78

200

Challenge:

Predict what the following pieces of code will do:

```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;
int main ()
ł
  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 << " ";
  }
  cout << "Blast off!!" << endl;</pre>
  return ∅;
}
    CSci 127 (Hunter)
                                   Lecture 12
```

July 2021 26 / 78

Sac

C++ Demo

```
//Another C++ program: Demonstrates loops
#include <iostream>
using namespace std;
int main ()
  int i,j;
  for (i = 0; i < 4; i++)
  £
      cout << "The world turned upside down...\n";</pre>
  3
  for (j = 10; j > 0; j - -)
  {
      cout << j << " ";
  3
  cout << "Blast off!!" << endl;</pre>
  return 0;
3
```

(Demo with onlinegdb)

CSci 127 (Hunter)

Lecture 12

3 July 2021 27 / 78

Sac

Definite loops

```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;
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 << " ";
  cout << "Blast off!!" << endl;</pre>
  return 0:
3
```

General format:

. . .

```
for (initialization; test; updateAction)
ί
    command1;
     command2;
     command3;
```

イロト イポト イヨト イヨト

3 July 2021 28 / 78

Sac

Challenge:

Predict what the following pieces of code will do:

```
//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++)
      cout << "*";
    cout << endl:
  }
  cout << "\n\n";</pre>
  for (i = size; i > 0; i--)
  {
    for (j = 0; j < i; j++)
      cout << "*":
    cout << endl;
  }
  return 0;
}
     CSci 127 (Hunter)
                                        Lecture 12
```

イロト イポト イヨト イヨト

э. July 2021 29 / 78

900

C++ Demo

```
//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++)</pre>
    for (j = 0; j < size; j++)
    cout << "*";
    cout << endl;
  3
  cout << "\n\n";</pre>
  for (i = size; i > 0; i - -)
  {
    for (j = 0; j < i; j++)
    cout << "*":
    cout << endl:
  3
  return 0;
3
```

(Demo with onlinegdb)

Sac

Challenge:

Predict what the following pieces of code will do:

```
//Growth example
#include <iostream>
using namespace std;
int main ()
{
  int population = 100;
  cout << "Year\tPopulation\n";</pre>
  for (int year = 0; year < 100; year= year+5)
  {
      cout << year << "\t" << population << "\n";</pre>
      population = population * 2;
  }
  return ∅;
}
```

Challenge:

Translate the C++ program into Python:

```
//Growth example
#include <iostream>
using namespace std;
int main ()
{
  int population = 100;
  cout << "Year\tPopulation\n";</pre>
  for (int year = 0; year < 100; year= year+5)
  {
      cout << year << "\t" << population << "\n";</pre>
      population = population * 2;
  }
  return 0;
```

CSci 127 (Hunter)

3 July 2021 32 / 78

200

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

• Efficient for systems programming.

```
//Another C++ program, demonstrating I/O & arithmetic
#include ciostreams
using namespace std;
int main O
{ float tg, lbs;
cout << "Enter kg; ";
int >> kg; ".2;
cout << endl << "Lbs; " << lbs << "\n\n";
}
enturn 0;
}
```

200

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

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

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

Sac

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

- 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 cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cin > kg;
  cin > kg;
  coat << end! << "lbs: " << lbs << "\n\n";
  return 0;
```

Sar

- 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 cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cin > kg;
  cin > kg;
  coat << end! << "lbs: " << lbs << "\n\n";
  return 0;
```

Sar

- 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
minclude cistoream.
using mamespace std;
int main O
{
  float kg, lbs;
  coat << "inter kg: ";
  cin > kg;
  cin > kg;
  coat << end! << "lbs: " << lbs << "\n\n";
  return 0;
```

Sac

| # | /Another C++ program, demonstrating I/O & arithmetic include <iostream></iostream> sing namespace std; |
|---------|--|
| ii { | nt main () float kg, 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, ...
```

To print:

900

| //Another C++ program, demonstrating I/O & arithmetic #include <iostream> using namespace std;</iostream> |
|--|
| int main () |
| <pre>{ float kg, lbs; }</pre> |
| 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!!";

Sac

| <pre>//Another C++ program, demonstrating I/O & arithmet #include <iostream> using namespace std;</iostream></pre> |
|---|
| <pre>int main O { font 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:

900

| <pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre> | et |
|---|----|
| 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;

Sac

| <pre>//Another C++ program, demonstrating 1/0 & arithme #include <iostream> using namespace std;</iostream></pre> |
|---|
| $\label{eq:constraint} \begin{array}{l} \text{int main } O \\ \{ \text{fott } k_0, \text{ bs}; \\ \text{cont} < < \text{"Enter } k_0; \; \text{"}; \\ \text{cin } > k_0; \\ \text{lbs } = k_0 \neq 2.2; \\ \text{cont} < < \text{end} < < \text{"Lbs}; \; \text{"} < \text{ lbs } < < \text{"\n'\n"}; \\ \text{return } 0; \\ \text{return } 0; \\ \end{array}$ |

- 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:

Sac

| <pre>//Another C++ program, #include <iostream> using namespace std;</iostream></pre> | demonstrating | 1/0 & | arithme |
|--|---------------|--------|---------|
| <pre>int main () { float kg, lbs; cout << "Enter kg: " cin >> kg; lbs = kg * 2.2; cout << endl << "Lbs return 0;</pre> | | "\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;

200

イロト 不得 トイヨト イヨト 二日

| <pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre> |
|--|
| <pre>int main () { for kg, lbs; cont << "Enter kg: "; cin >> kg; lbs = kg * 2.2; cont << 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: cin >> num;
- To use those I/O functions: #include <iostream>
 - using namespace std;
- Definite loops:

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

| <pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre> |
|---|
| int main () |
| { |
| float kg, lbs; |
| cout << "Enter kg: "; |
| cin >> kg; |
| lbs = kg * 2.2; |
| <pre>cout << endl << "Lbs: " << 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!!";
- 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>

イロト イポト イヨト イヨト

Sac

| <pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre> |
|---|
| <pre>int main () { float kg, lbs; cout << "Enter kg: "; cin >> kg; lbs = kg * 2.2; }</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!!";
- 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>

イロト 不得 トイヨト イヨト 二日

• Blocks of code uses '{' and '}'.

| <pre>//Another C++ program, demonstrating I/O & arithme #include <iostream> using namespace std;</iostream></pre> |
|---|
| int main () { |
| float kg, lbs; |
| cout << "Enter kg: "; |
| cin >> kg; |
| lbs = kg * 2.2; |
| cout and "the: " the "\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++) $\{\ldots\}$

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

Lecture 12

July 2021 33 / 78

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

Today's Topics

```
//Acaba C++ program, demonstrating 1/0 & arithmetic using namespace std; int main () finat kg, lbs; cout < "finat kg, lbs; cout < "lbs < "white it is the set of the set
```

- Recap: Incrementer Design Challenge
- C++: Basic Format & Variables
- I/O and Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python
- More Info on the Final Exam

▲ E ● E • ○ Q ○ July 2021 34 / 78

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: ";</pre>
    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 \leq 2004)
        cout << "Millennial":</pre>
    }
    else
        cout << "TBD":
    return 0:
```

```
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>
```

イロト イポト イヨト イヨト

CSci 127 (Hunter)

Lecture 12

3 July 2021 35 / 78

C++ 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 ∅;
}
```

Sac

Conditionals

General format:



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

CSci 127 (Hunter)

3 July 2021 37 / 78

Sar

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

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

and (&&)

| in1 | | in2 | returns: |
|-------|----|-------|----------|
| False | && | False | False |
| False | && | True | False |
| True | && | False | False |
| True | && | 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 | False | False |
| False | True | True |
| True | False | True |
| True | True | True |

CSci 127 (Hunter)

July 2021 38 / 78

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 12

July 2021 38 / 78

Today's Topics

```
//Acctive C++ program, demonstrating L/O & arithmetic
include costermespace std;
int main O
f Totat kg, lbs;
coat c+ "Inter kg; ";
coat c+ of There kg; ";
coat c+ of The
```

- Recap: Incrementer Design Challenge
- C++: Basic Format & Variables
- I/O and Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python
- More Info on the Final Exam

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

CSci 127 (Hunter)

July 2021 40 / 78

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

```
///Mile Growth Example
finclude <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;
    }
    return 0;
}</pre>
```

(Demo with onlinegdb)

Indefinite Loops: while

```
///While Growth Example
#include <iostream>
using namespace std;
int main ()
Ł
  int population = 100;
  int year = 0;
  cout << "Year\tPopulation\n";</pre>
 while(population < 1000)</pre>
  {
   cout << vear << "\t\t" << population << "\n";</pre>
    population = population * 2:
    year++:
  return 0:
```

General format:

```
while ( logical expression )
ł
```

command1; command2; command3;

3 July 2021 42 / 78

Sac

Challenge:

Predict what the following piece of code will do:

```
//Demonstrates loops
#include <iostream>
using namespace std;
int main ()
ł
  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;
  }
  cout << "You entered: "</pre>
        << num << ".\n";
  return 0;
}
```

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

```
//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 ∅;
```

▲□▶ ▲□▶ ▲ヨ▶ ▲ヨ▶ ヨ - のの⊙

Indefinite Loops: while

```
//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;
3
```

General format:

```
while (logical expression)
```

command1; command2; command3;

. . .

Sac

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;
}
                                        イロト イポト イヨト イヨト
```

C++ Demo

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

CSci 127 (Hunter)

3 July 2021 47 / 78

200

<ロト <回ト < 回ト < 回ト

Indefinite Loops: do-while

```
//Demonstrates do-while loops
#include <iostream>
                                             General format:
using namespace std:
int main ()
                                             do
                                              {
  int num;
  do
                                                    command1;
  {
      cout << "Enter an even number: ";</pre>
                                                    command2;
      cin >> num:
                                                    command3;
  } while (num % 2 != 0);
                                                    . . .
  cout << "You entered: "</pre>
                                              } while ( logical expression );
       << num << ".\n";
  return 0:
}
```

Lecture 12

3 July 2021 48 / 78

Sac

Today's Topics

```
//Acother C++ program, demonstrating L/O & arithmetic include costerminespace std; ist tain O ( for a std), ist cost c, "Inter Ng: "; cost c, "Inter Ng: "; cost c, "Inter Ng: "; cost c, end c, "Lbs: " \ll lbs \ll "\n\n"; "return 0; "return 0;
```

- Recap: Incrementer Design Challenge
- C++: Basic Format & Variables
- I/O and Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python
- More Info on the Final Exam

▲ 王 シ へ C
 July 2021 49 / 78

• I/O:

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

#inity monospore txd;

int main ()

int
```

990

• I/O: cin >> ...;

Jac.

• I/O: cin >> ...; & cout << ...;

200

• I/O: cin >> ...; & cout << ...;

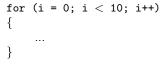
Definite loops:

```
//Another C++ program: Demonstrates loops
#include <iostream>
using namespace std;
int main ()
 int i,j;
  for (i = 0; i < 4; i++)
     cout << "The world turned upside down...\n";</pre>
  3
  for (j = 10; j > 0; j - -)
     cout << j << " ";
  cout << "Blast off!!" << endl:
  return 0;
```

Sac

• I/O: cin >> ...; & cout << ...;

Definite loops:



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

```
int main ()
 int i,j;
 for (i = 0; i < 4; i + )
     cout << "The world turned upside down...\n";</pre>
 3
 for (j = 10; j > 0; j - -)
 ł
     cout << j << " ";
 cout << "Blast off!!" << endl:
 return 0;
```

Sac

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

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

```
int main ()
 int i,j;
 for (i = 0; i < 4; i + )
     cout << "The world turned upside down...\n";</pre>
 3
  for (j = 10; j > 0; j - -)
 ł
     cout << j << " ";
 cout << "Blast off!!" << endl:
 return 0;
```

Sac

```
//Another C++ program; Demonstrates loops
#include <iostream>
using namespace std;
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 << " ";
 cout << "Blast off!!" << endl:
 return 0;
```

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

Sac

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

3 July 2021 50 / 78

Sac

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

```
I/O: cin >> ...; & cout << ...;
</pre>
Definite loops:
  for (i = 0; i < 10; i++)
        ...
Conditionals:
  if (logical expression)
Indefinite loops:
  while (logical expression)
        ...
                      イロト イポト イヨト イヨト
                                                Sac
                                             3
     Lecture 12
                                      July 2021
                                                50 / 78
```

CSci 127 (Hunter)

• *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>
```

CSci 127 (Hunter)

< ロト < 同ト < ヨト < ヨ

• *Rewrite this program in C++:*

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

イロト 不良 トイヨト イヨト ヨー のくや

• *Rewrite this program in C++:*

```
for i in range(2017, 2000, -2):
    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;
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)
```

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~

```
• Rewrite this program in C++:
```

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

```
#include <iostream>
using namespace std;
int main()
{
   for (int i = 2017; i > 2000; i=i-2)
    {
      cout << "Year is " << i << endl;
   }
}</pre>
```

CSci 127 (Hunter)

200

```
• 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;
   }
   return 0;
}</pre>
```

CSci 127 (Hunter)

July 2021 52 / 78

• 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>
```

Sac

イロト イロト イヨト イ

• Rewrite this program in Python:

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

CSci 127 (Hunter)

A (10) + (10)

• Rewrite this program in Python:

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

CSci 127 (Hunter)

July 2021 53 / 78

- 4 伊ト 4 ヨト 4

Sac

```
• 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).

CSci 127 (Hunter)

July 2021 54 / 78

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

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

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

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

CSci 127 (Hunter)

July 2021 55 / 78

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~

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

▲□▶ ▲□▶ ▲ヨ▶ ▲ヨ▶ ヨ - のの⊙

```
• 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")
```

CSci 127 (Hunter)

July 2021 56 / 78

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~

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

▲□▶ ▲□▶ ▲ヨ▶ ▲ヨ▶ ヨ - のの⊙

```
• 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")
```

CSci 127 (Hunter)

July 2021 57 / 78

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~

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

▲□▶ ▲□▶ ▲ヨ▶ ▲ヨ▶ ヨ - のの⊙

```
• 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")
```

CSci 127 (Hunter)

July 2021 58 / 78

```
• 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")
```

CSci 127 (Hunter)

July 2021 59 / 78

```
• 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")
```

CSci 127 (Hunter)

July 2021 60 / 78

```
• 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 \setminus
      (not (year % 100 == 0) or (year % 400 == 0)):
       print("Leap!!")
  print("Year")
```

```
year = 2016
if TRUE:
    print("Leap!!")
print("Year")
```

```
Prints: Leap!
Year
                                                ◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○
CSci 127 (Hunter)
                                Lecture 12
                                                                July 2021
```

61 / 78

 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;

200

 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()

200

 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;
```

Sac

• 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>

Sac

• 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;

Sac

• 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>

Sac

• 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";

Challenge: Conditionals in Python & C++

• 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"; } else

イロト イポト イヨト イヨト

Challenge: Conditionals in Python & C++

• 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"; else cout << "Cheaper to buy 7-day unlimited card.\n";

イロト イポト イヨト イヨト

Challenge: Conditionals in Python & C++

```
• 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";
    else
      cout << "Cheaper to buy 7-day unlimited card.\n";
    return 0;
                                              イロト イポト イヨト イヨト
                                                                = nac
  CSci 127 (Hunter)
                               Lecture 12
                                                           July 2021
                                                                    62 / 78
```

• 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 12

July 2021 63 / 78

<ロト < 団ト < 団ト < 団ト < 団ト = 三</p>

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

999

イロト イポト イヨト イヨト 二日

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

s = ""

999

イロト イポト イヨト イヨト 二日

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

s = "" while s == "":

Sac

イロト イポト イヨト イヨト 一日

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

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

200

イロト イポト イヨト イヨト 一日

• 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)
```

▲□▶ ▲□▶ ▲ヨ▶ ▲ヨ▶ ヨ - のの⊙

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.

200

イロト 不得 トイヨト イヨト 二日

• 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;

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

• 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()

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

• 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;

200

イロト 不得 トイヨト イヨト 二日

• 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)

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

• 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>
```

CSci 127 (Hunter)

July 2021 64 / 78

イロト 不得 トイヨト イヨト ヨー のくや

• 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;
```

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ − ∽ Q (~

• 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;
 }
}

= nac

イロト イポト イヨト イヨト

• 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; イロト イポト イヨト イヨト ≡ ∽ar CSci 127 (Hunter) Lecture 12 July 2021 64 / 78

Today's Topics



- Recap: Incrementer Design Challenge
- C++: Basic Format & Variables
- I/O and Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++
- Recap: C++ & Python
- More Info on the Final Exam

900

イロト イポト イヨト イヨト

 Although the exam is online, we still suggest you prepare 1 piece of 8.5" x 11" paper.

Sac

- Although the exam is online, we still suggest you prepare 1 piece of 8.5" x 11" paper.
 - ▶ With notes, examples, programs: what will help you on the exam.

Sac

- Although the exam is online, we still suggest you prepare 1 piece of 8.5" x 11" paper.
 - ▶ With notes, examples, programs: what will help you on the exam.
 - Best if you design/write yours, it's excellent way to study.

Sac

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

- Although the exam is online, we still suggest you prepare 1 piece of 8.5" x 11" paper.
 - ▶ With notes, examples, programs: what will help you on the exam.
 - Best if you design/write yours, it's excellent way to study.
- The exam format:

Sac

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

- Although the exam is online, we still suggest you prepare 1 piece of 8.5" x 11" paper.
 - ▶ With notes, examples, programs: what will help you on the exam.
 - Best if you design/write yours, it's excellent way to study.
- The exam format:
 - Like a long quiz on Gradecope, need to scroll down a lot.

イロト イポト イヨト イヨト

- Although the exam is online, we still suggest you prepare 1 piece of 8.5" x 11" paper.
 - ▶ With notes, examples, programs: what will help you on the exam.
 - ► Best if you design/write yours, it's excellent way to study.
- The exam format:
 - ► Like a long quiz on Gradecope, need to scroll down a lot.
 - Many questions that roughly correspond to the 10 parts on old paper finals.

イロト イポト イヨト イヨト

- Although the exam is online, we still suggest you prepare 1 piece of 8.5" x 11" paper.
 - ▶ With notes, examples, programs: what will help you on the exam.
 - ► Best if you design/write yours, it's excellent way to study.
- The exam format:
 - ► Like a long quiz on Gradecope, need to scroll down a lot.
 - Many questions that roughly correspond to the 10 parts on old paper finals.
 - Questions based on course topics, and are variations on the programming assignments, lab exercises, and lecture design challenges.

イロト イポト イヨト イヨト 二日

- Although the exam is online, we still suggest you prepare 1 piece of 8.5" x 11" paper.
 - ▶ With notes, examples, programs: what will help you on the exam.
 - ► Best if you design/write yours, it's excellent way to study.
- The exam format:
 - ► Like a long quiz on Gradecope, need to scroll down a lot.
 - Many questions that roughly correspond to the 10 parts on old paper finals.
 - Questions based on course topics, and are variations on the programming assignments, lab exercises, and lecture design challenges.
 - Style of questions: short answer, fill in the program (one line of code per box), multiple choice, select all, replace value, modify program, translate & write complete programs.

- Although the exam is online, we still suggest you prepare 1 piece of 8.5" x 11" paper.
 - ▶ With notes, examples, programs: what will help you on the exam.
 - ► Best if you design/write yours, it's excellent way to study.
- The exam format:
 - ► Like a long quiz on Gradecope, need to scroll down a lot.
 - Many questions that roughly correspond to the 10 parts on old paper finals.
 - Questions based on course topics, and are variations on the programming assignments, lab exercises, and lecture design challenges.
 - Style of questions: short answer, fill in the program (one line of code per box), multiple choice, select all, replace value, modify program, translate & write complete programs.
- Past exams available on webpage (includes answer keys).

イロト イポト イヨト イヨト 二日



• Emphasis of this course is on analytic reasoning and problem solving.

CSci 127 (Hunter)

Lecture 12

3 July 2021 67 / 78

590

▲ロト ▲圖ト ▲目下



- Emphasis of this course is on analytic reasoning and problem solving.
- The best way to prepare to do problems (reading & watching videos can clarify but not replace problem solving).

Image: A match a ma

July 2021 67 / 78



- Emphasis of this course is on analytic reasoning and problem solving.
- The best way to prepare to do problems (reading & watching videos can clarify but not replace problem solving).

イロト イロト イヨト イ

• Repeat, while there are past exams:

July 2021 67 / 78



- Emphasis of this course is on analytic reasoning and problem solving.
- The best way to prepare to do problems (reading & watching videos can clarify but not replace problem solving).
- Repeat, while there are past exams:
 - Choose a past exam (see webpage).

イロト イロト イヨト イ



- Emphasis of this course is on analytic reasoning and problem solving.
- The best way to prepare to do problems (reading & watching videos can clarify but not replace problem solving).
- Repeat, while there are past exams:
 - Choose a past exam (see webpage).
 - With only a note sheet, work through in 1 hour (half the time).

イロト イポト イヨト イ

July 2021 67 / 78



- Emphasis of this course is on analytic reasoning and problem solving.
- The best way to prepare to do problems (reading & watching videos can clarify but not replace problem solving).
- Repeat, while there are past exams:
 - Choose a past exam (see webpage).
 - ► With only a note sheet, work through in 1 hour (half the time).

イロト イポト イヨト イ

Grade yourself (answers on webpage).



- Emphasis of this course is on analytic reasoning and problem solving.
- The best way to prepare to do problems (reading & watching videos can clarify but not replace problem solving).
- Repeat, while there are past exams:
 - Choose a past exam (see webpage).
 - ► With only a note sheet, work through in 1 hour (half the time).
 - Grade yourself (answers on webpage).
 - Ask about those that don't make sense.

イロト イロト イヨト イ

July 2021 67 / 78



- Emphasis of this course is on analytic reasoning and problem solving.
- The best way to prepare to do problems (reading & watching videos can clarify but not replace problem solving).
- Repeat, while there are past exams:
 - Choose a past exam (see webpage).
 - ► With only a note sheet, work through in 1 hour (half the time).
 - Grade yourself (answers on webpage).
 - Ask about those that don't make sense.

イロト イポト イヨト イヨ

 Rewrite answers & organize by type/question number.

How to Prepare



- Emphasis of this course is on analytic reasoning and problem solving.
- The best way to prepare to do problems (reading & watching videos can clarify but not replace problem solving).
- Repeat, while there are past exams:
 - Choose a past exam (see webpage).
 - ► With only a note sheet, work through in 1 hour (half the time).
 - Grade yourself (answers on webpage).
 - Ask about those that don't make sense.
 - Rewrite answers & organize by type/question number.
 - Adjust/rewrite note sheet to include what you wished you had.

You will get credit for you answers only if:

990

You will get credit for you answers only if:

• Your answer uses language constructs that were covered in the course.

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

You will get credit for you answers **only if**:

- Your answer uses language constructs that were covered in the course.
- Even if your answer is correct, it will get 0 points if the method was not covered in this course.

200

You will get credit for you answers only if:

- Your answer uses language constructs that were covered in the course.
- Even if your answer is correct, it will get 0 points if the method was not covered in this course.
- Your answer is not obviously copy/pasted from a website.

You will get credit for you answers only if:

- Your answer uses language constructs that were covered in the course.
- Even if your answer is correct, it will get 0 points if the method was not covered in this course.
- Your answer is not obviously copy/pasted from a website.
- Your answer is not oddly identically to that of another student or is the answer for another version of the exam.

You will get credit for you answers only if:

- Your answer uses language constructs that were covered in the course.
- Even if your answer is correct, it will get 0 points if the method was not covered in this course.
- Your answer is not obviously copy/pasted from a website.
- Your answer is not oddly identically to that of another student or is the answer for another version of the exam.

All acts of academic dishonesty will be reported to the Office of Academic and Student Affairs and will result in a 0 grade on the exam.

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that takes a weight in kilograms and returns the weight in pounds.

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that takes a weight in kilograms and returns the weight in pounds.

def kg2lbs(kg):

...
return(lbs)

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that takes a weight in kilograms and returns the weight in pounds.

```
def kg2lbs(kg)
    lbs = kg * 2.2
    return(lbs)
```

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that takes a string and returns its length.

Sac

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that takes a string and returns its length.

```
def sLength(str):
```

```
...
return(length)
```

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that takes a string and returns its length.

```
def sLength(str):
    length = len(str)
    return(length)
```

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

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that, given a DataFrame, returns the minimal value in the "Manhattan" column.

イロト 不得 トイヨト イヨト 二日

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that, given a DataFrame, returns the minimal value in the "Manhattan" column.

def getMin(df):
 ...
 return(min)

CSci 127 (Hunter)

Lecture 12

July 2021 73 / 78

For each question below, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that, given a DataFrame, returns the minimal value in the "Manhattan" column.

```
def getMin(df):
    min = df['Manhattan'].min()
    return(min)
```

イロト 不得 トイヨト イヨト 二日

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that takes a whole number and returns the corresponding binary number as a string.

イロト 不得 トイヨト イヨト 二日

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that takes a whole number and returns the corresponding binary number as a string.

def num2bin(num):

```
...
return(bin)
```

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that takes a whole number and returns the corresponding binary number as a string.

```
def num2bin(num):
    binStr = ""
    while (num > 0):
        #Divide by 2, and add the remainder to the string
        r = num %2
        binString = str(r) + binStr
        num = num / 2
    return(binStr)
```

CSci 127 (Hunter)

July 2021 76 / 78

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that computes the total monthly payment when given the initial loan amount, annual interest rate, number of years of the loan.

For each question, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that computes the total monthly payment when given the initial loan amount, annual interest rate, number of years of the loan.

def computePayment(loan,rate,year):

....
return(payment)

For each question below, write the function header (name & inputs) and return values (often called the Application Programming Interface (API)):

• Write a function that computes the total monthly payment when given the initial loan amount, annual interest rate, number of years of the loan.

def computePayment(loan,rate,year):
 (Some formula for payment)
 return(payment)

<ロト < 回 > < 回 > < 回 > < 回 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < 三 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □