Top-Down Design

CSC 161: The Art of Programming
Prof. Henry Kautz
9/16/2009

Today's Lesson

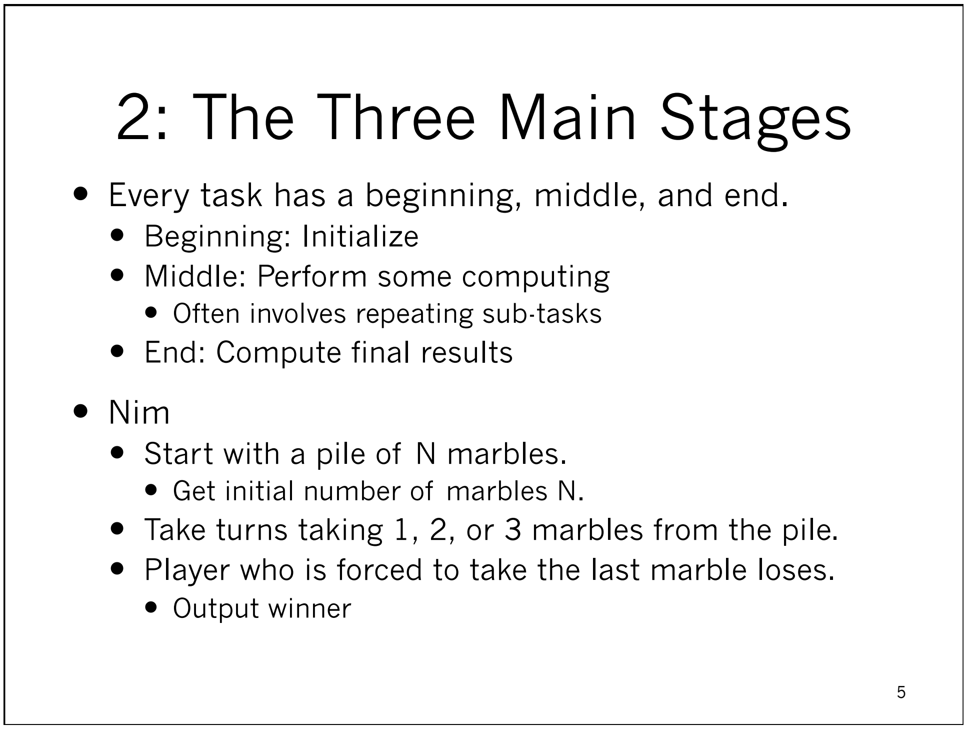
- Don't panic!
- Top-down design
- Summary of the Elements of Python
- Don't panic!

Nim

- Two player game.
- Start with a pile of N marbles.
- Take turns taking 1, 2, or 3 marbles from the pile.
- Player who is forced to take the last marble loses.

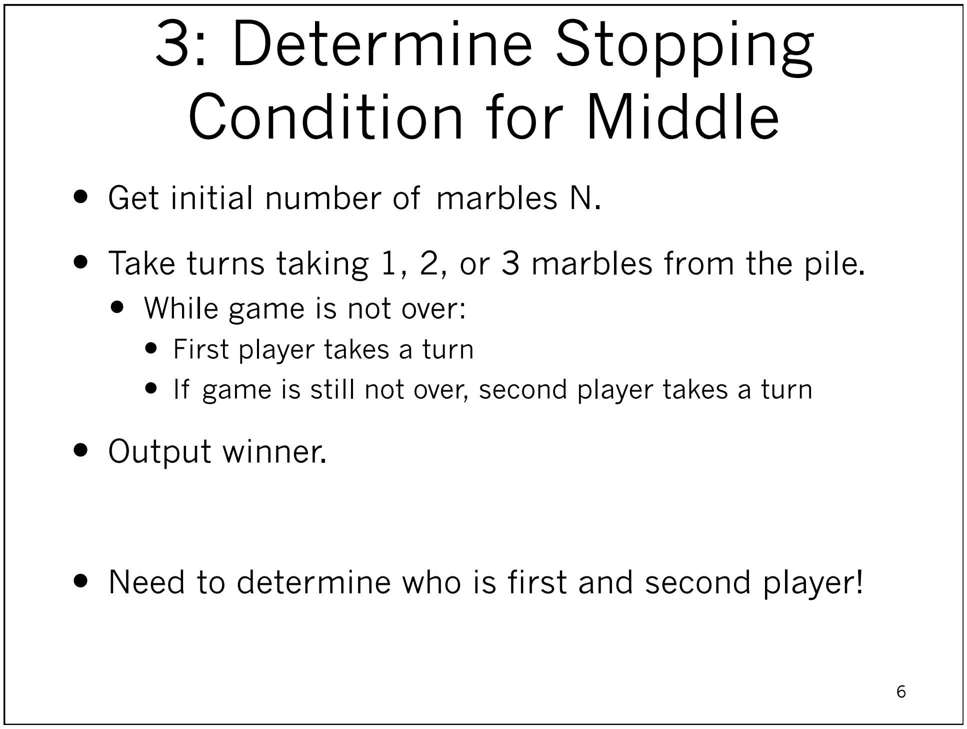
1: State the Task

- Play a game of Nim against a human opponent, using the rules:
  - Start with a pile of N marbles.
  - Take turns taking 1, 2, or 3 marbles from the pile.
  - Player who is forced to take the last marble loses.



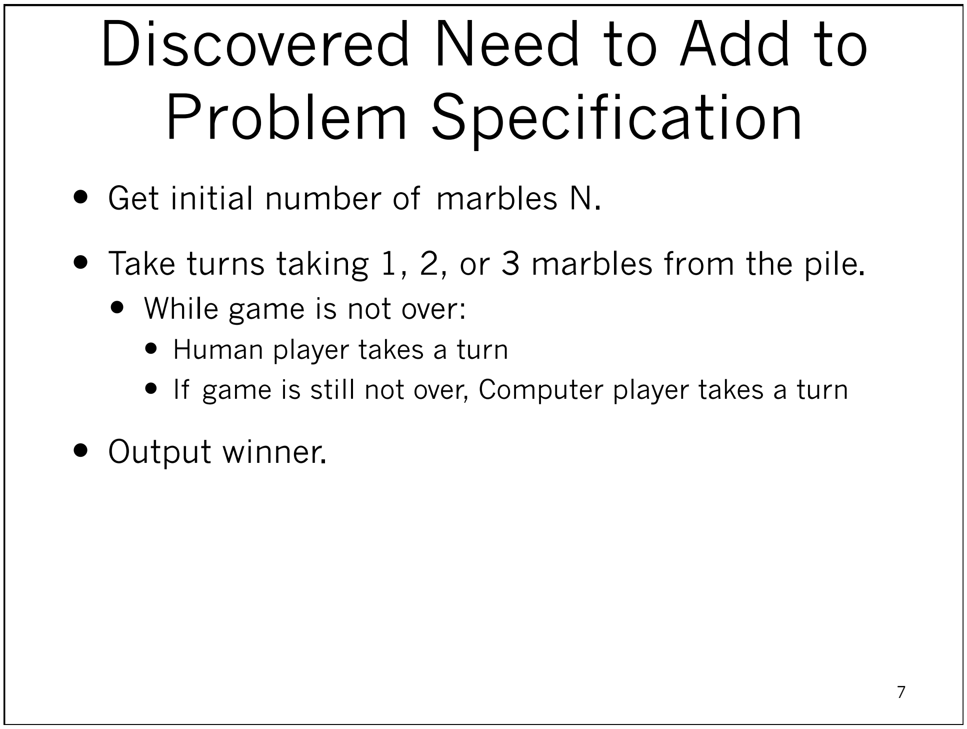
2: The Three Main Stages

- Every task has a beginning, middle, and end.
  - Beginning: Initialize
  - Middle: Perform some computing
    - Often involves repeating sub-tasks
  - End: Compute final results
- Nim
  - Start with a pile of N marbles.
    - Get initial number of marbles N.
  - Take turns taking 1, 2, or 3 marbles from the pile.
  - Player who is forced to take the last marble loses.
    - Output winner



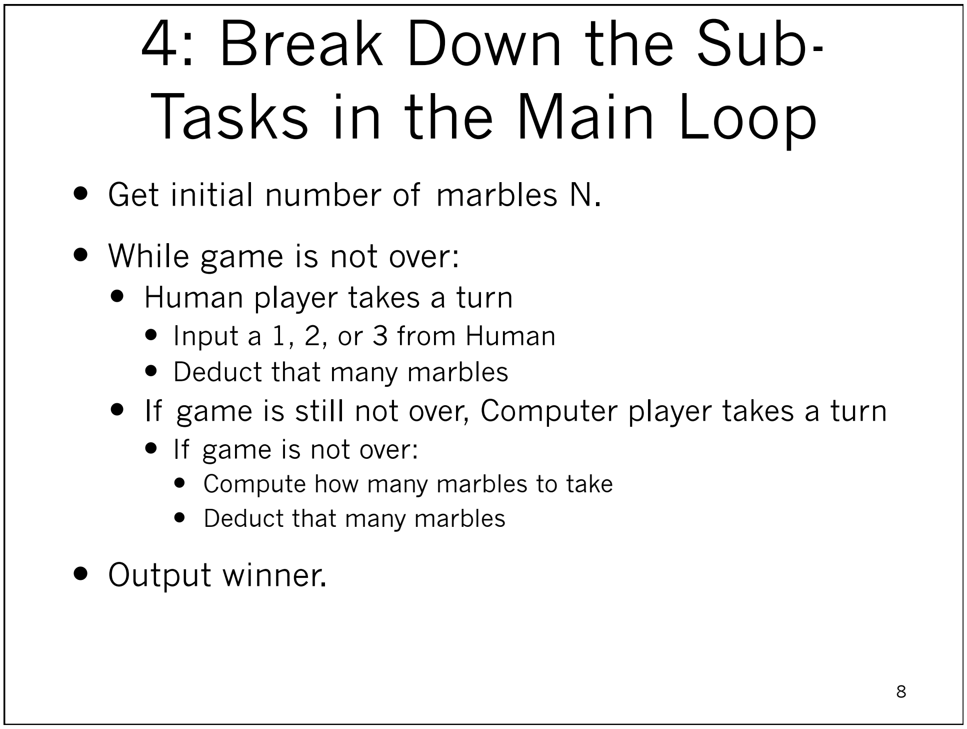
3: Determine Stopping Condition for Middle

- Get initial number of marbles N.
- Take turns taking 1, 2, or 3 marbles from the pile.
  - While game is not over:
    - First player takes a turn
    - If game is still not over, second player takes a turn
- Output winner.
- Need to determine who is first and second player!



Discovered Need to Add to Problem Specification

- Get initial number of marbles N.
- Take turns taking 1, 2, or 3 marbles from the pile.
  - While game is not over:
    - Human player takes a turn
    - If game is still not over, Computer player takes a turn
- Output winner.



4: Break Down the Sub-Tasks in the Main Loop

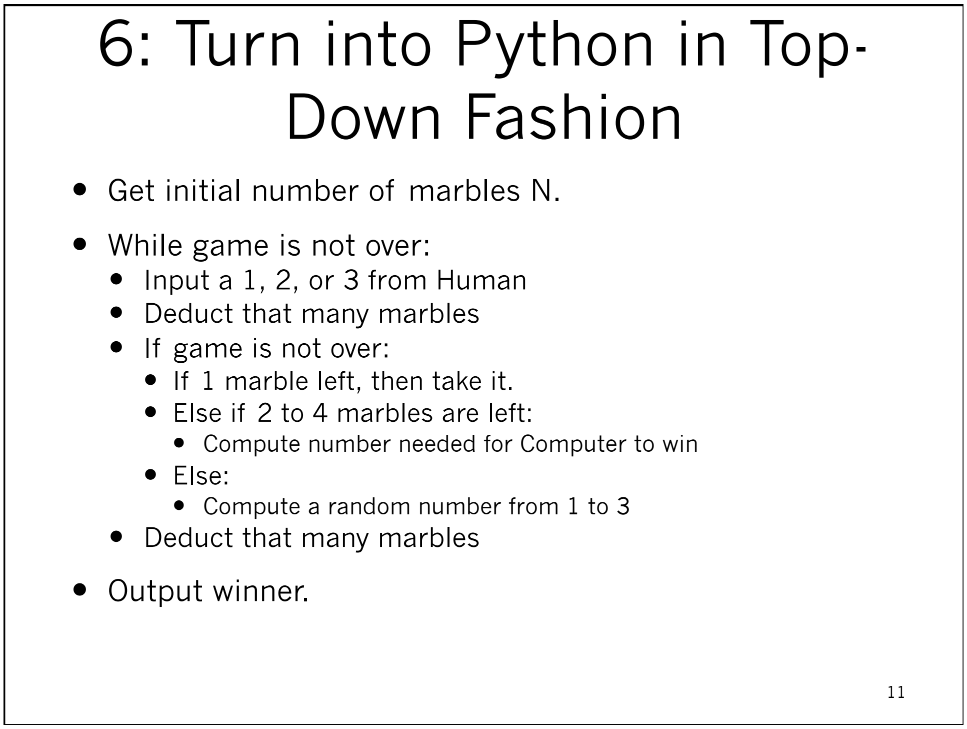
- Get initial number of marbles N.
- While game is not over:
  - Human player takes a turn
    - Input a 1, 2, or 3 from Human
    - Deduct that many marbles
  - If game is still not over, Computer player takes a turn
    - If game is not over:
      - Compute how many marbles to take
      - Deduct that many marbles
- Output winner.

5: Find Sub-Tasks that Need to be Broken Down Further

- Get initial number of marbles N.
- While game is not over:
  - Human player takes a turn
    - Input a 1, 2, or 3 from Human
    - Deduct that many marbles
  - If game is still not over, Computer player takes a turn
    - If game is not over:
      - Compute how many marbles to take
      - Deduct that many marbles
- Output winner.

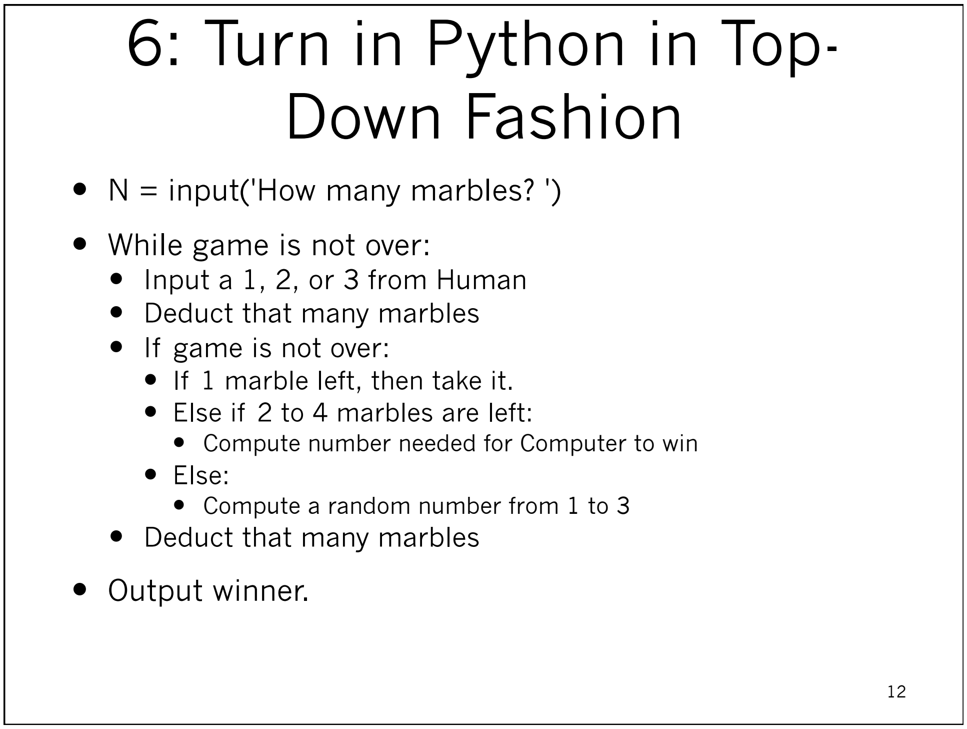
5: Find Sub-Tasks that Need to be Broken Down Further

- Get initial number of marbles N.
- While game is not over:
  - Input a 1, 2, or 3 from Human
  - Deduct that many marbles
  - If game is not over:
    - Compute how many marbles to take
      - If 1 marble left, then take it
      - Else if 2 to 4 marbles are left:
        - Compute number needed for Computer to win
      - Else:
        - Compute a random number from 1 to 3
    - Deduct that many marbles
- Output winner.



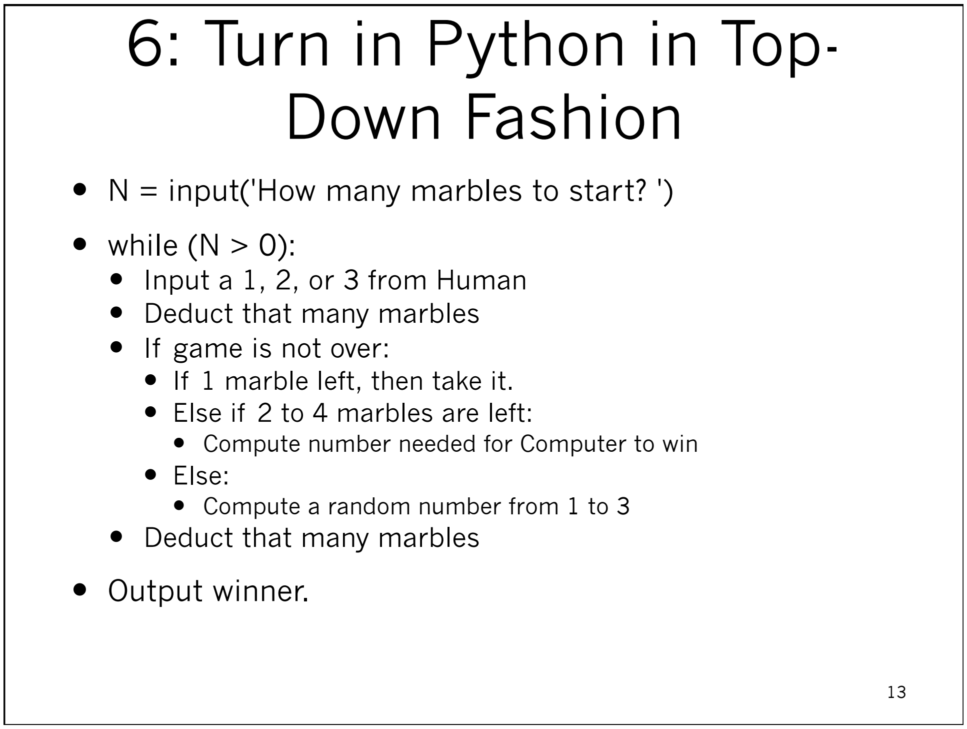
6: Turn into Python in Top-Down Fashion

- Get initial number of marbles N.
- While game is not over:
  - Input a 1, 2, or 3 from Human
  - Deduct that many marbles
  - If game is not over:
    - If 1 marble left, then take it.
    - Else if 2 to 4 marbles are left:
      - Compute number needed for Computer to win
    - Else:
      - Compute a random number from 1 to 3
  - Deduct that many marbles
- Output winner.



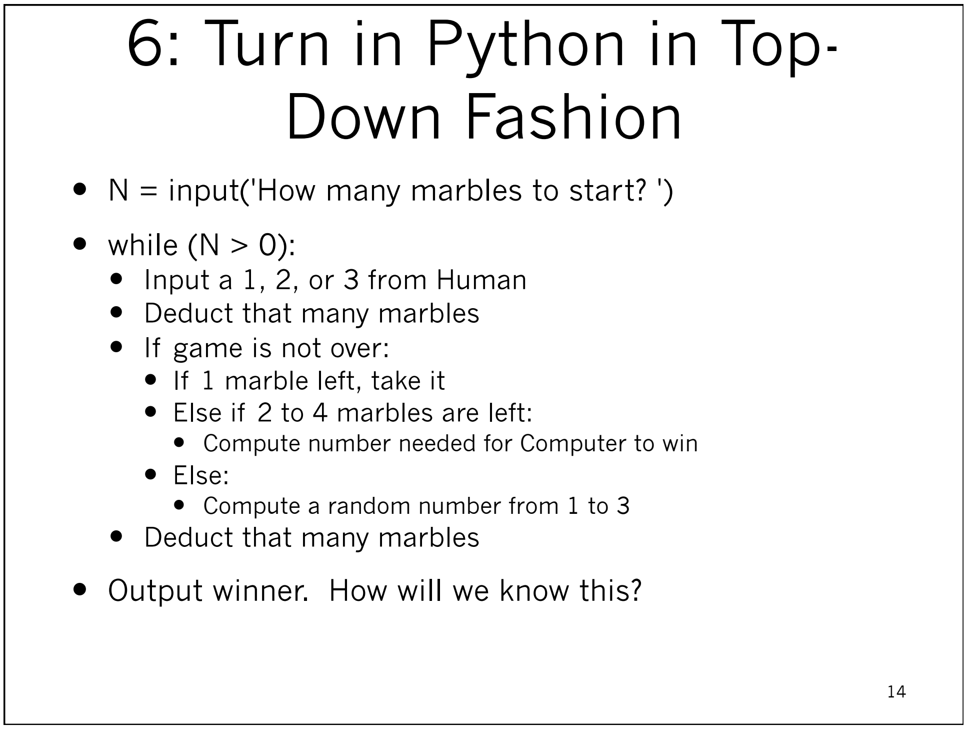
6: Turn in Python in Top-Down Fashion

- N = input('How many marbles? ')
- While game is not over:
  - Input a 1, 2, or 3 from Human
  - Deduct that many marbles
  - If game is not over:
    - If 1 marble left, then take it.
    - Else if 2 to 4 marbles are left:
      - Compute number needed for Computer to win
    - Else:
      - Compute a random number from 1 to 3
  - Deduct that many marbles
- Output winner.



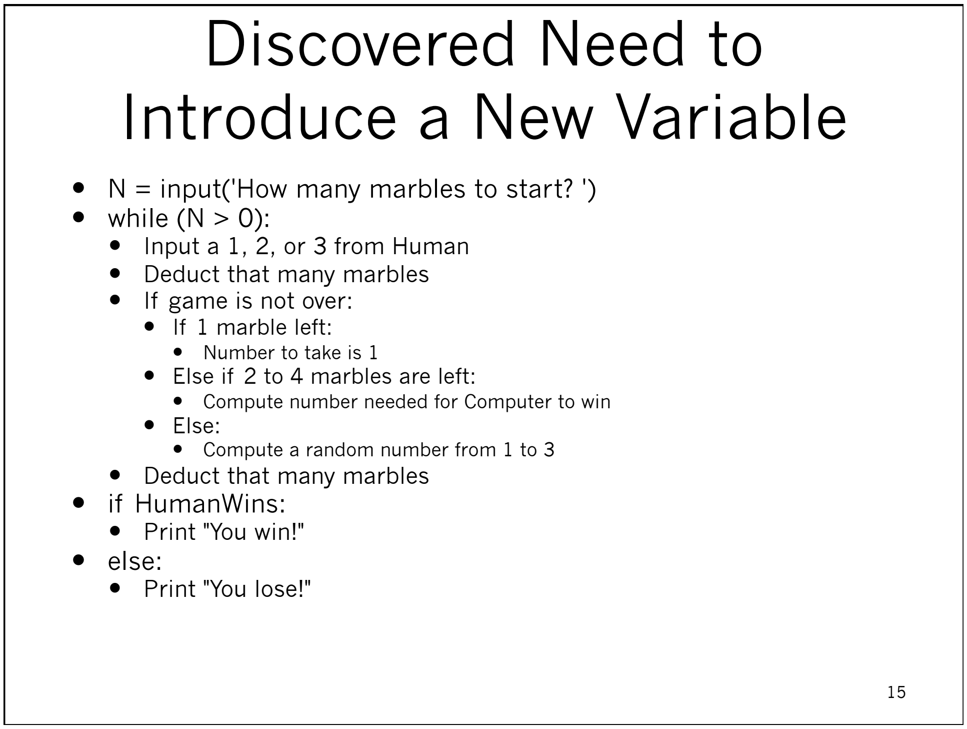
6: Turn in Python in Top-Down Fashion

- N = input('How many marbles to start? ')
- while (N > 0):
  - Input a 1, 2, or 3 from Human
  - Deduct that many marbles
  - If game is not over:
    - If 1 marble left, then take it.
    - Else if 2 to 4 marbles are left:
      - Compute number needed for Computer to win
    - Else:
      - Compute a random number from 1 to 3
  - Deduct that many marbles
- Output winner.



6: Turn in Python in Top-Down Fashion

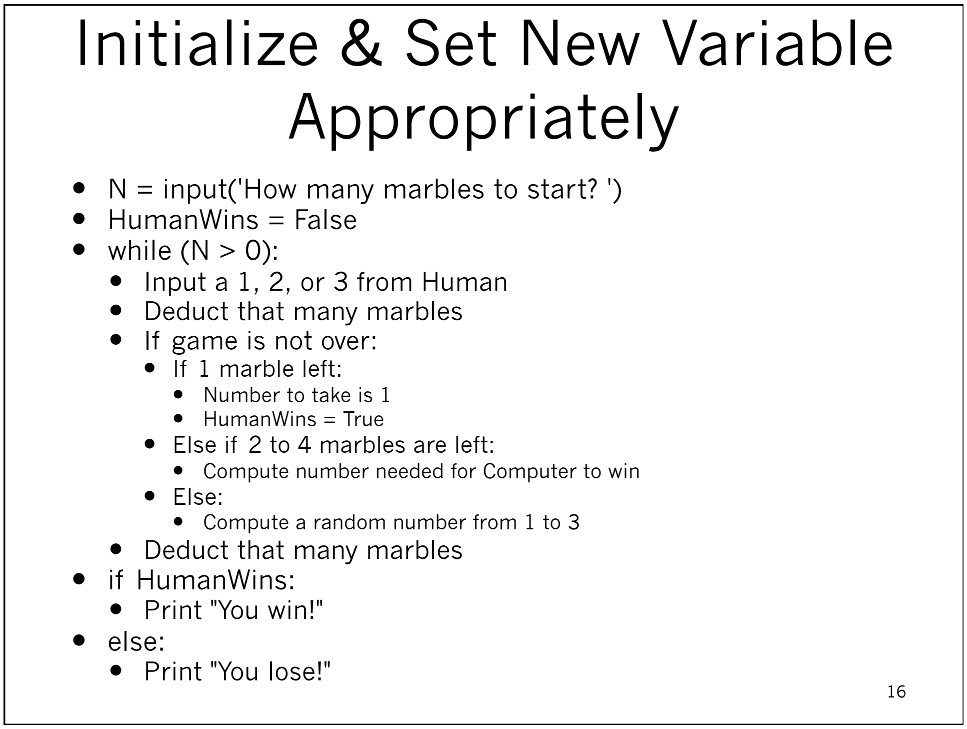
- N = input('How many marbles to start? ')
- while (N > 0):
  - Input a 1, 2, or 3 from Human
  - Deduct that many marbles
  - If game is not over:
    - If 1 marble left, take it
    - Else if 2 to 4 marbles are left:
      - Compute number needed for Computer to win
    - Else:
      - Compute a random number from 1 to 3
  - Deduct that many marbles
- Output winner. How will we know this?



Discovered Need to Introduce a New Variable

- N = input('How many marbles to start? ')
- while (N > 0):
  - Input a 1, 2, or 3 from Human
  - Deduct that many marbles
  - If game is not over:
    - If 1 marble left:
      - Number to take is 1
    - Else if 2 to 4 marbles are left:
      - Compute number needed for Computer to win
    - Else:
      - Compute a random number from 1 to 3
  - Deduct that many marbles
- if HumanWins:
  - Print "You win!"
- else:
  - Print "You lose!"

15



Initialize & Set New Variable Appropriately

- N = input('How many marbles to start? ')
- HumanWins = False
- while (N > 0):
  - Input a 1, 2, or 3 from Human
  - Deduct that many marbles
  - If game is not over:
    - If 1 marble left:
      - Number to take is 1
      - HumanWins = True
    - Else if 2 to 4 marbles are left:
      - Compute number needed for Computer to win
    - Else:
      - Compute a random number from 1 to 3
  - Deduct that many marbles
- if HumanWins:
  - Print "You win!"
- else:
  - Print "You lose!"

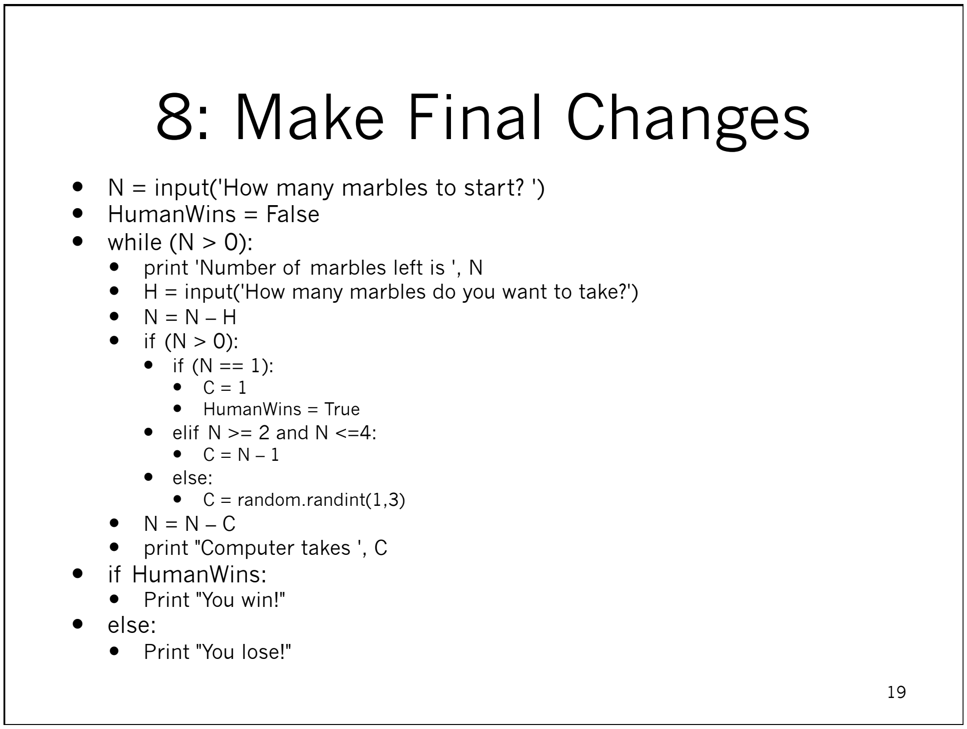
16

Finish Turning into Python

- N = input('How many marbles to start? ')
- HumanWins = False
- while (N > 0):
  - H = input('How many marbles do you want to take?')
  - N = N – H
  - if (N > 0):
    - if (N == 1):
      - C = 1
      - HumanWins = True
    - elif N >= 2 and N <=4:
      - C = N – 1
    - else:
      - C = random.randint(1,3)
  - N = N – C
- if HumanWins:
  - Print "You win!"
- else:
  - Print "You lose!"

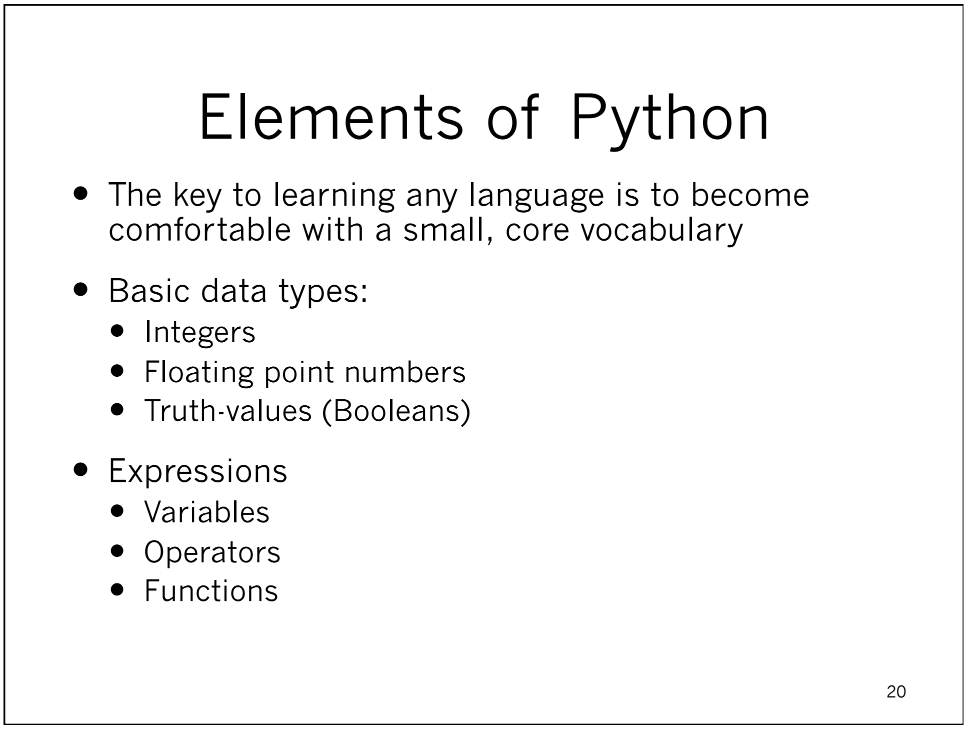
7: Review for Correctness and Completeness

- Program should tell player how many marbles are left
- Program should tell player how many marbles it is taking



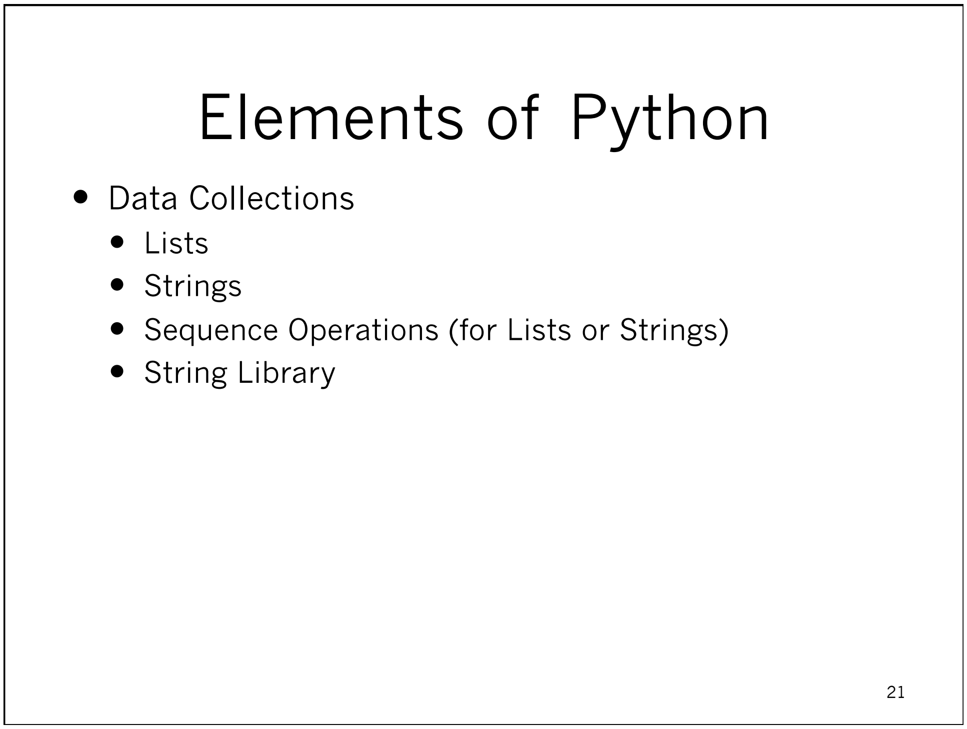
8: Make Final Changes

- N = input('How many marbles to start? ')
- HumanWins = False
- while (N > 0):
  - print 'Number of marbles left is ', N
  - H = input('How many marbles do you want to take?')
  - N = N – H
  - if (N > 0):
    - if (N == 1):
      - C = 1
      - HumanWins = True
    - elif N >= 2 and N <=4:
      - C = N – 1
    - else:
      - C = random.randint(1,3)
  - N = N – C
  - print "Computer takes ', C
- if HumanWins:
  - Print "You win!"
- else:
  - Print "You lose!"



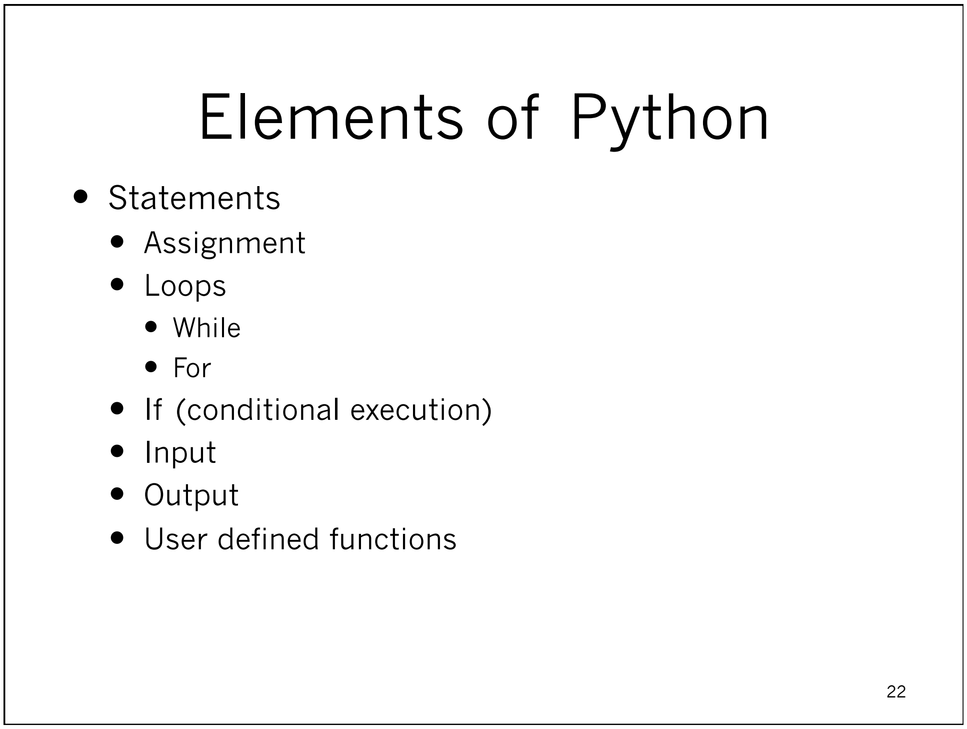
Elements of Python

- The key to learning any language is to become comfortable with a small, core vocabulary
- Basic data types:
  - Integers
  - Floating point numbers
  - Truth-values (Booleans)
- Expressions
  - Variables
  - Operators
  - Functions



Elements of Python

- Data Collections
  - Lists
  - Strings
  - Sequence Operations (for Lists or Strings)
  - String Library



Elements of Python

- Statements
  - Assignment
  - Loops
    - While
    - For
  - If (conditional execution)
  - Input
  - Output
  - User defined functions