CMPT 120
Introduction To Computing Science And Programming I

Pseudocode

Summer 2012

Instructor: Hassan Khosravi

Guessing game

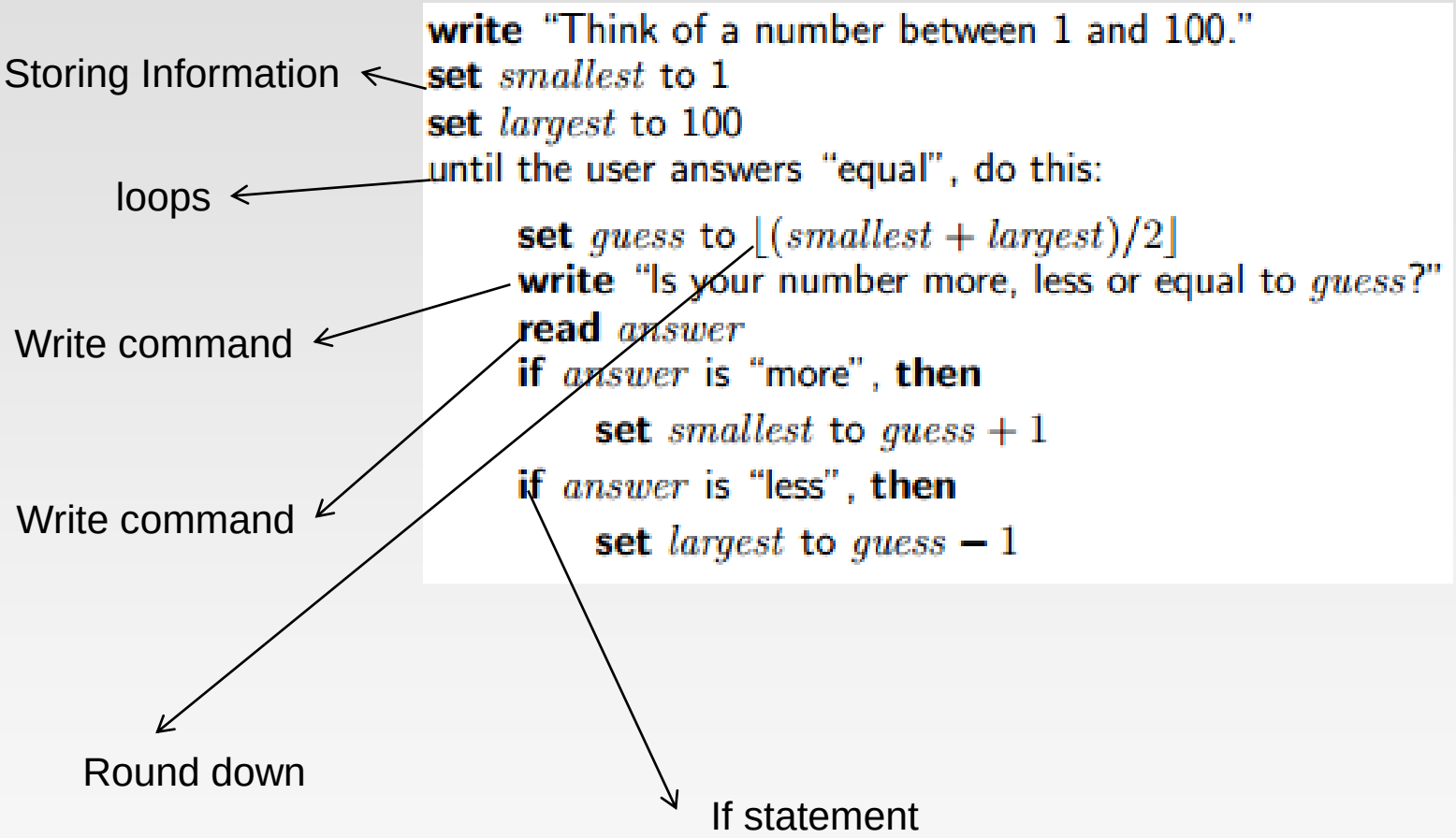
1. Tell the user to pick a secret number between 1 and 100.
2. The smallest possible number is 1; the largest possible is 100.
3. Make a guess that is halfway between the smallest and largest (round down if necessary).
4. Ask the user if your guess is too large, too small or correct.
5. If they say you're correct, you win and the game is over.
6. If they say your guess is too small, the smallest possible number is now the guess plus one.
7. If they say your guess is too large, the largest possible number is now the guess minus one.
8. Unless you guessed correctly, go back to step 3.

- How many guess are required if your number is 33?
- Can you think of a number where the algorithm in Figure 1.2 will make 7 guesses?
- Can you think of a number where the algorithm in Figure 1.2 will make 8 guesses?
- What is “legitimate input” for this algorithm? What happens if the user enters something else?

Pseudocode

- you need a way to describe the algorithms that you are going to implement. This is often done with pseudocode

- “pseudo-” means “almost” → Pseudocode is almost code


write "Think of a number between 1 and 100."
set smallest to 1
set largest to 100
until the user answers "equal", do this:
set guess to ⌊(smallest + largest)/2⌋
write "Is your number more, less or equal to guess?"
read answer
if answer is "more", then
set smallest to guess + 1
if answer is "less", then
set largest to guess − 1
Storing Information
loops
Write command
Write command
Round down
If statement

Pseudocode

- Storing Information:
  - Set variable to value
    - Set x to 1
    - Set y to “hello”
    - Set z to True
- Write command: writes whatever is inside “” in the terminal
  - Write “Hello”
  - Write x
    - Outputs the value of x
- Read Command: sets the value of the variable to what is read from the terminal
  - Read x

Example

- Read two numbers and return their average

1. Write "Enter two numbers"
2. read first_number
3. read second_number
4. set average to (first_number + second_number)/ 2
5. write "The average is"
6. write average

Example

- Read the height of a person in meters and output their height in inches
  - 1 meter = 39.37 inch

1. write"Enter your height in meters"
2. read height_in_meters
3. set height_in_inches to height_in_meters * 39.37
4. Write "Your height is"
5. write height_in_inches
6. write "inches tall"

Example

- Read the height of a person in meters and output their height in feet
  - 1 feet = 12 inch

1. write"Enter your height in meters"
2. read height_in_meters
3. set height_in_feet to height_in_meters * 39.37 /12
4. Write "Your height is"
5. write height_in_feet
6. write "feet tall"

- Round down
  - set y to $\lfloor X \rfloor$
    - If x is 2.3 then y=2
    - If x -2.3 then y = -3

- Round up
  - Set y to $\lceil X \rceil$
    - If x is 2.3 then y=3
    - If x is -2.3 then y=-2

Example

- Read the height of a person in meters and output their height in feet and inches

1. write"Enter your height in meters"
2. read **height_in_meters**
3. set **height_in_inches** to **height_in_meters** * 39.37
4. set **feet** to **height_in_inches/12**
5. round down feet to nearest integer
6. set **inches** to **height_in_inches** - (**feet** *12)
7. Round down **inches**
8. write "you are"
9. write **feet**, **inches**
10. write "tall“

- Let’s try 1.80
  - height in meters: 1.80
  - height in inches 70.866
  - height in feet 5.09 → 5
  - inches 10.866 → 10
  - you are 5 10 tall

If statement

- The most common way to make decisions is by using the if statement.

- If Command
  - If statement then
    - Do some stuff
  - Else
    - Do some other stuff
- The else part is optional and can be omitted.

  - Read X
  - If X < 2 then
    - Write X “is smaller than two”
  - Else
    - Write X “is bigger or equal to two”

- Read a number, if it is odd then write odd, if it is even write it is even

1. write "Enter a number"
2. Read number
3. set number to number/2
4. set number_roundup to round up number
5. set number_rounddown to round down number
6. if number_roundup = number_rounddown then print
7. write"even"
8. else
9. write "odd"

- What do you think of Pseudocode and the examples

- A: They are very easy and the pace of the class is slow
- B: They are understandable and the pace is right
- C: I am finding the examples a little hard to follow. The pace of the class is too fast
- D: I just don’t get Pseudocode

- Read a number between 1 and 10. try to guess it? If you guess it correctly say you win, else say you lose.

1. write "Enter a number between 1 to 10"
2. Read number
3. set guess to random value between 0 to 10
4. if number = guess then
5. 	write "Wow"
6. else
7. 	write "Nope, wrong answer"

Question

What would be the output if you input 13

1. Write “How old are you”
2. if age <= 2 then
3. write “you fly for free”
4. Else if 2 < age < 13 then
5. write “you pay the kids rate‘”
6. Else then
7. write 'you pay regular adult fare’

A: No output

B: you fly for free

C: you pay the kids rate

D: you pay regular adult fare

E: You get an error

The Value for age is undefined, so it doesn’t make sense to compare it. You will get an error

- Read three numbers and return their maximum

1. Write "Enter three numbers"
2. read num1,num2,num3
3. if num1>num2 then
4. 	if num1 > num3 then
5. 		write num1
6. 	else
7. 		write num3
8. else
9. 	if num2 > num3 then
10. 		write num2
11. 	else
12. 		write num3

Definite Iteration: for loops

- We need to be able to execute the same code several times
- The for loop
  - The for loop can be used when you know ahead of time how many times you want to execute some code
  - For a value i equal to each number from 1 to n:
    - Statement regarding I
- Compute the factorial for a input value
  - n! = 1 × 2 × 3 × · · · × (n − 1) × n .

1. write “Enter a nonnegative integer:”
2. read n
3. set factorial to 1
4. for i equal to each number from 1 to n:
5. set factorial to factorial × i
6. write factorial

Examples

- Read a value n and count from 1 to n

- write "Enter a number"
- read n
- for i equal to each number from 1 to n\
- write i

Examples

- Compute the average of 10 numbers

1. set sum to 0
2. for i equal to each number from 1 to 10
3. read num
4. set sum to sum + num
5. average = sum/10
6. write average

Examples

- Read a value and determine whether it is prime or not

1. write "read number"
2. read n
3. set prime to True

4. for i from 1 to roundup(n/2)
5. 	if remainder of n/i is 0
6. 		set prime to False
7. if prime = True
8. 	write "your number is prime"
9. if prime = False
10. 	write "your number is not prime"

Examples

- Compute the first n values in the Fibonacci series.
  - Fibonacci series are the numbers in the following integer sequence:
    - 0,1,1,2,3,5,…
    - $F_n = F_{n-1} + F_{n-2}$
    - $F_0 = 0$, $F_1 = 1$

1. set smaller to 0
2. set bigger to 1
3. write smaller
4. write bigger
5. do this for i equal to each number from 1 to n-2\
6. newnum = smaller + bigger
7. write newnum
8. smaller = bigger
9. bigger= newnum

Trace the code to see what the 8th number in the series would be

- A: 10
- B:8
- C:13
- D:21
- E:none of the above

Indefinite Iteration: while loops

- If you don't know how many times you want the loop body to execute, the for loop is hard to work with.
  - While a statement holds do
    - Some stuff

write "Think of a number between 1 and 100."
set *smallest* to 1
set *largest* to 100
until the user answers "equal", do this:

    set *guess* to $\lfloor (smallest + largest)/2 \rfloor$
    write "Is your number more, less or equal to *guess*?"
    read *answer*
    if *answer* is "more", **then**
        set *smallest* to $guess + 1$
    if *answer* is "less", **then**
        set *largest* to $guess - 1$

- You could write this with a for loop if you analyze it carefully ☺

Example

- Calculate the sum of some positive numbers, when you are finished with your numbers give -1 as your number to terminate the program

1. sum of some positive numbers
2. write "Enter some numbers"
3. set sum to 0
4. read number
5. while number >0
6. 	sum = sum +num
7. 	read number
8. write "sum of numbers are"
9. write sum

Example

- Enter a value n. Find the smallest number k where k^2 is bigger than n

1. write"Enter a value"
2. read n
3. k = 0
4. while k*k =< n
5. K=K+1
6. write k

If we put num1 as 13 and num2 as 3, what would be the output

1. write"Enter two numbers"
2. read num1,num2
3. num3=0
4. while num1>= num2
5. num1 = num1 - num2
6. num3 = num3+1
7. write num3, num1

- A: num3 = 2, num1=2
- B: num3 = 4, num1=1
- C: num3=3, num1 = 5
- D: num3=0, num1=0
- E none of the Above