General Procedure for Writing a While Loop

Class Exercise

• Using our postage function from Wednesday, write a program that takes as input the weight (in ounces) of a package and calculates the required postage amount to mail it. Allow the user to continue to calculate postage for as long as they’d like.

• Assume the total postage is calculated as follows:
  • A flat rate of $0.90 plus $0.20 for every ounce over 1

Loop Practice

Write a while loop that will compute the sum of the first $n$ positive odd numbers. For example, if $n$ is 5, you should compute $1 + 3 + 5 + 7 + 9$.

Announcements

• Reminders
  — Program 4 due Sunday, October 1st by 11:55pm
While Loop General Algorithm

while some condition is true:
    # Do this code
    # Something here should modify the condition above
    #ATM Machine - keeps asking for your PIN until
    # you enter the right one
    pin = int(input("What is your PIN? "))
    while pin != 1234:
        print("Wrong PIN")
        pin = int(input("What is your PIN? "))

You try

Write a while loop that
- asks the user for 2 names
- compares the names
- outputs the name that comes first alphabetically.
- asks the user if they want to do it again at the end of the loop.
- keeps the loop going until the user replies ‘no’.

Modification as a class

Edit the previous loop so that the loop ends when the user enters ‘STOP’ for the first name. That is, the user no longer explicitly has to answer the question, ‘Do you want to keep going?’

You try

Write a loop to continuously ask the user for an integer from the keyboard. Stop looping only when they type in a number between 1 and 10.
Pseudocode

• an informal way of writing algorithms for humans to read (not computers!)
• Illustrates the logic of an algorithm, but omits details that people can fill in automatically.
• You get to make it up as you go along, as long as you (and other people) can easily understand it.

• Instead of saying
  name = input("What is your name? ")
• Pseudocode might use a line that says
  name = ask user for name

• Instead of saying
  if x >= 0 and x <= 100:
    print("%.2f" % x)
• Pseudocode might use a line that says
  if x is between 0 and 100:
    print x with 2 decimal places

The point is to get your ideas down on paper quickly, so you can worry about the programming details and exact syntax later.

To write any while loop:
1. Write out pseudocode for what the loop does, explicitly repeating lines until you've repeated the code at least twice.
2. Include an "if" statement in your code that will be True if you want the loop to keep going.
3. Make sure the code repeats the "if" statement at least twice.
4. Find the statements between consecutive "if" statements. These statements will become the body of the loop.
5. The "if" test will become the "while" test.
6. If there's anything before the first "if" test, it will go immediately before the while loop (outside of the body).
Pseudocode for Name Comparison Program

```python
name1 = ask for first name
if name1 is not STOP, then keep going
name2 = ask for second name
if name1 < name2, print out name1, else print name2
name1 = ask for first name
if name1 is not STOP, then keep going
name2 = ask for second name
if name1 < name2, print out name1, else print name2
```

Generating Random Numbers

- Random number are useful in a lot of programming tasks
- `random` module includes library functions for working with random numbers
- `randint` function: generates a random number in the range provided by the arguments

```python
number = random.randint(1, 100)
print(f'[random.randint(1, 100)]
```

Generating Random Numbers

Figure 6.2: A statement that calls the random function

Using Random Numbers Example

```python
# This program displays a random number
# in the range of 1 through 10.
import random

def main():
    # Get a random number.
    number = random.randint(1, 10)
    # Display the number.
    print(f'random.randint(1, 10)')
    # Call the main function.
main()
```
Pseudocode for Guess the Number Game

number = generate a random number between 1 and 100

if number does not equal guess, then keep going
    if guess is greater than number, tell user this
else if guess is less than number, tell user this

    guess = ask user to guess a number between 1 and 100

    if number does not equal guess, then keep going
        if guess is greater than number, tell user this
    else if guess is less than number, tell user this

    guess = ask user to guess a number between 1 and 100

    if number does not equal guess, then keep going
        if guess is greater than number, tell user this
    else if guess is less than number, tell user this

    . . .

    When user guesses correct number, tell them so

Sample Output

Guess a number between 1 and 100. 87
Your guess was too high.
Please try again.
Your guess was too high.
Please try again.

Your guess was too high.
Please try again.

You guessed right. Great work!