ENDG 233 – Programming with Data

Repetition Structures-Loops

Date: Oct. 11 – 17

Schedule for Week 6

UNIVERSITY OF CALGARY

- Review
- Examples on Loops
- Portfolio Project #2

Review: Loops

UNIVERSITY OF CALGARY

- **Loop** – is a program construct that repeatedly executes the loop's statements(known as the loop body) while the loop's expression is true, and breaks out of the loop once the expression is false.
- **Iteration** – each time through loop statements is called an iteration.
- Python has two primitive loop commands:
  - while loop
  - for loop

Review: While Loops

- **While loop** – is a construct that repeatedly executes an intended block of code (known as loop body) as long as loop's expression is true
- **Example**:

```
i = 1                    # initialize i
while i < 5:             # execute as long as condition is true
    print(i)
    i += 1               # increment I          i = i+1
```

Output:
1
2
3
4

Review: While Loops

- **Break statement** – can stop the loop even if while is true
- **Example**:

```
i = 1                       # initialize i
while i < 5:                # execute as long as condition is true
    print(i)
    if i == 3:              # exit the loop when i is 3
        break
    i += 1                  # increment i
```

Output:
1
2
3

Review: While Loops

- **Continue statement** – can stop current iteration and continue with next
- **Example**:

```
i = 1                       # initialize I
print(i)
print('loop number is', i)
while i < 5:                # execute as long as condition is true
    i += 1                  # increment i
    print(i)
    if i == 3:              # jump to the beginning of the loop when i is 3
        continue
    print('loop number is', i)
```

```
1
loop number is 1
2
loop number is 2
3
4
loop number is 4
5
loop number is 5
```

Review: While Loops

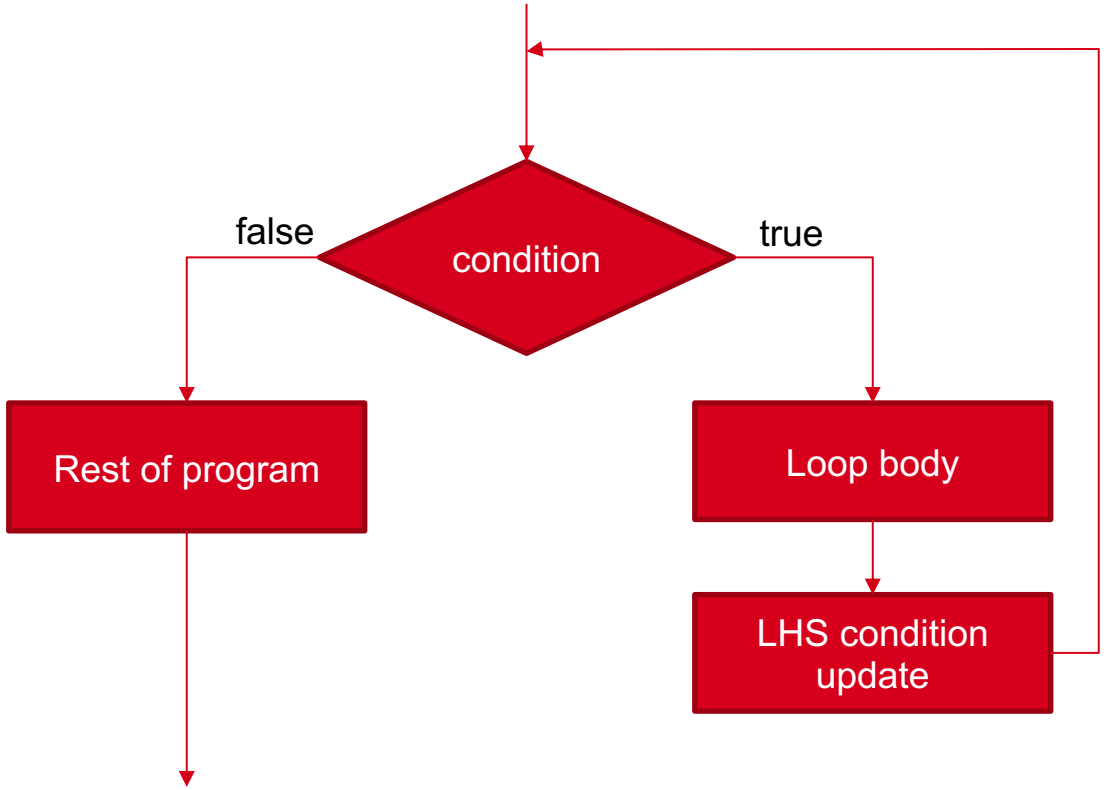
- **Else statement** – you can use else statement with while
- **Example**:

```
i = 1                       # initialize i
while i < 5:                # execute as long as condition is true
    print(i)
    i += 1                  # increment I
else:
    print("i is no longer less than 5")
```

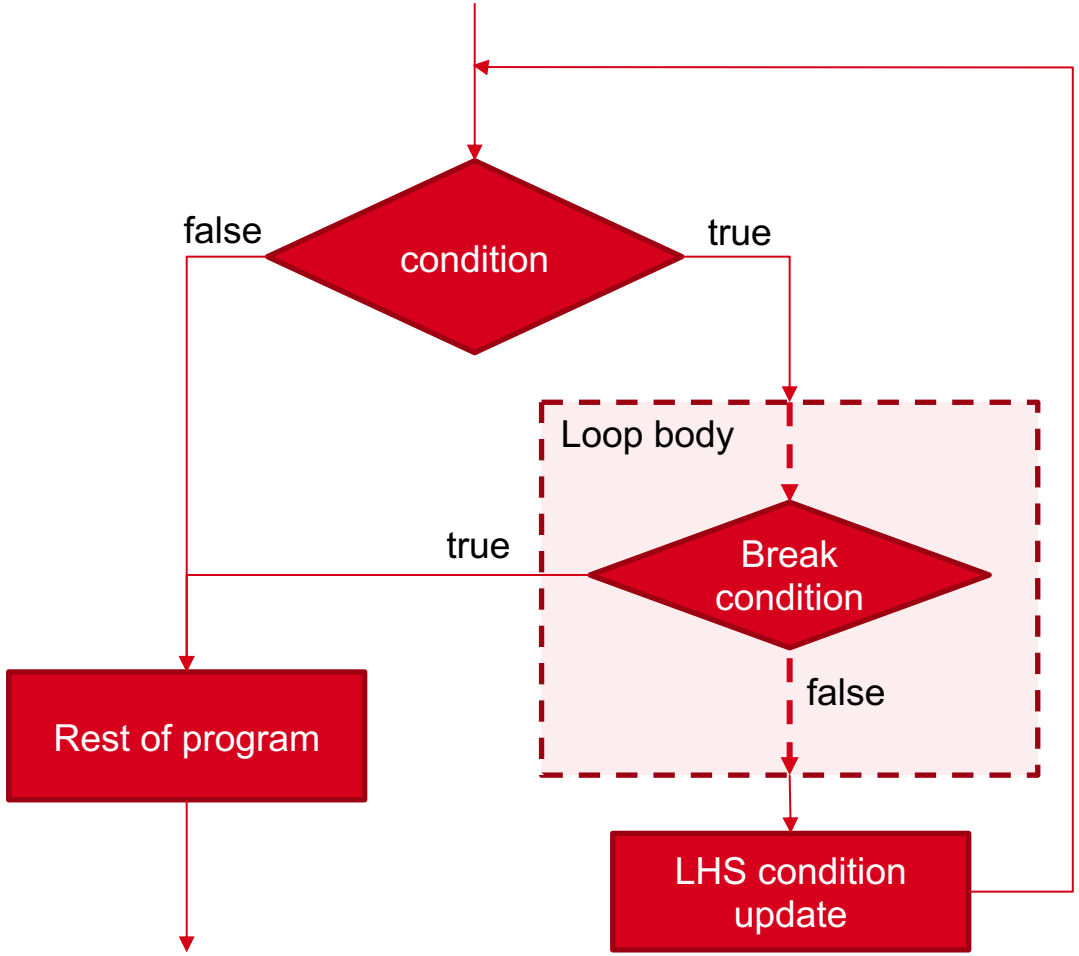
Output:

1
2
3
4
i is no longer less than 5

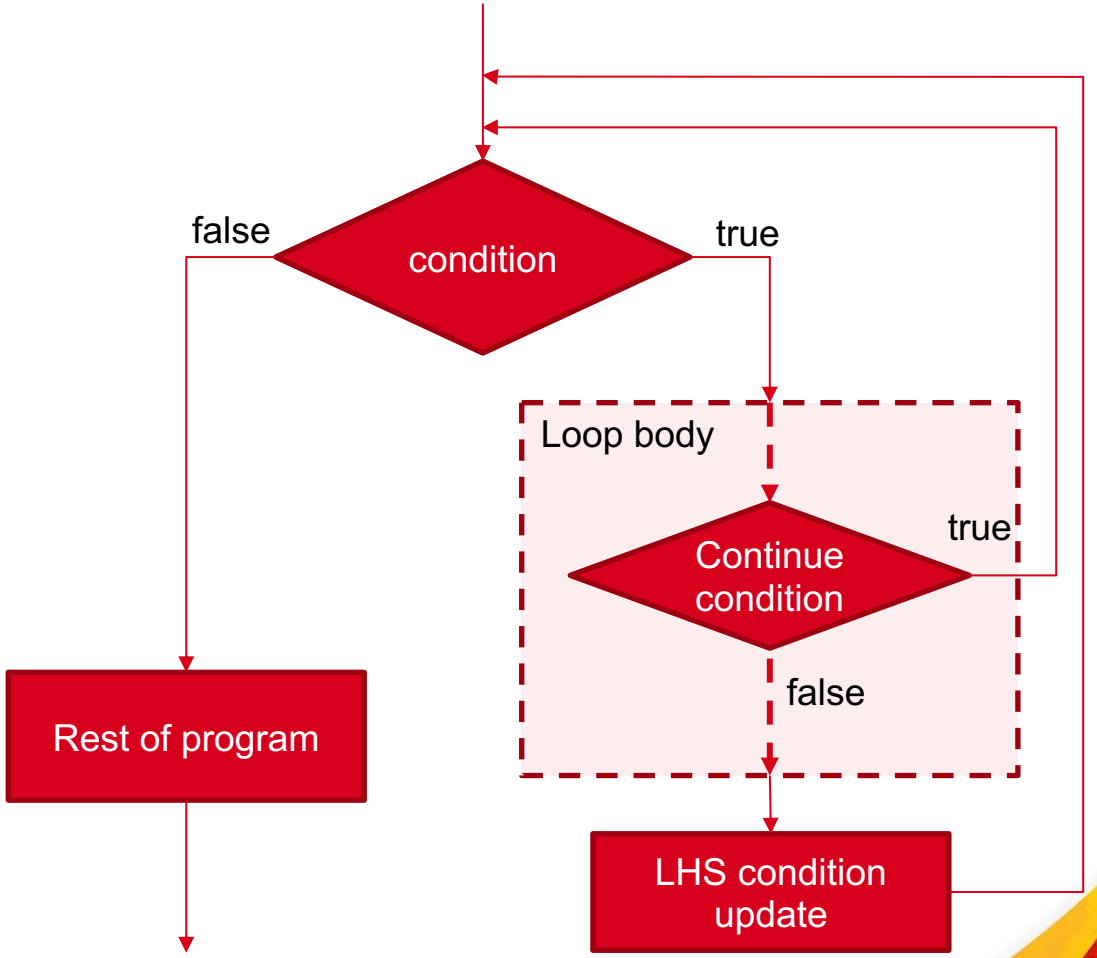
While in flowcharts



While in flowcharts

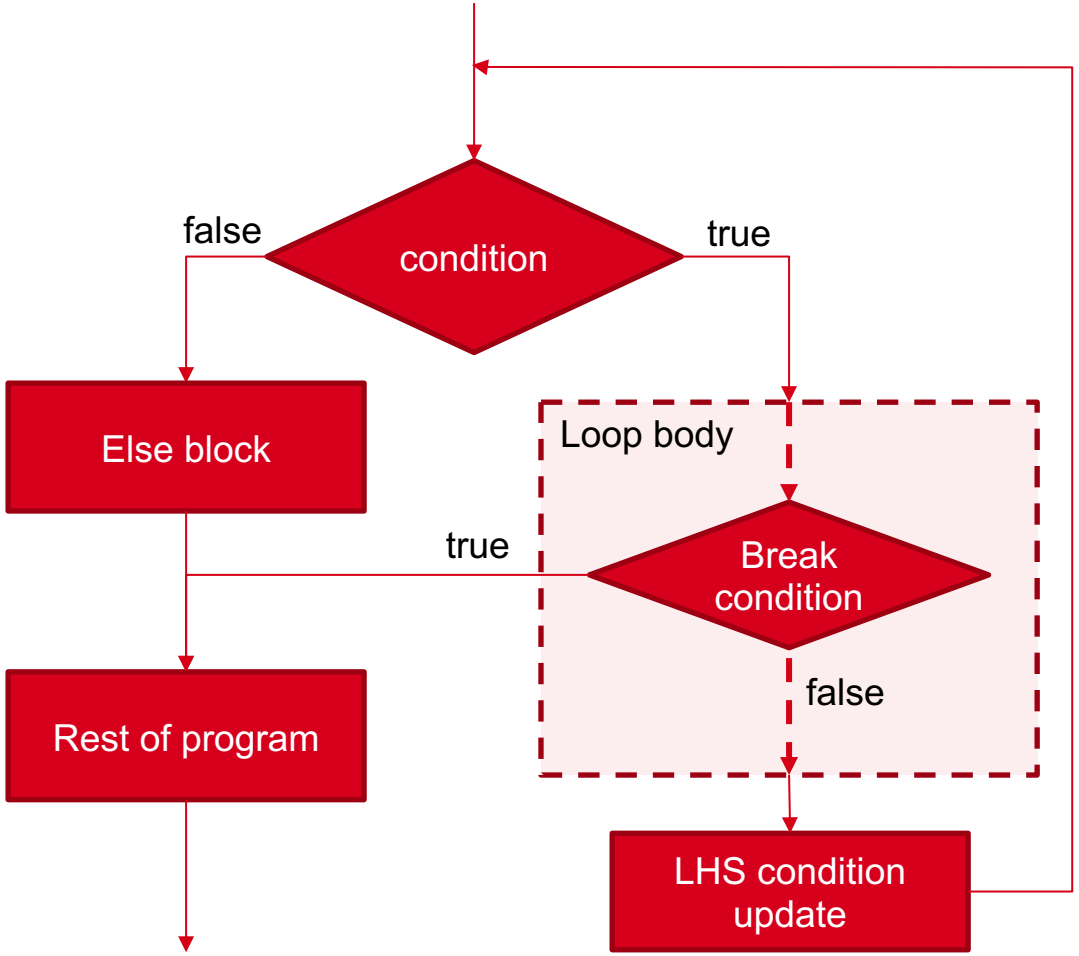


Breaks in flowcharts



Continue in flowcharts

While else in flowcharts



Review: For Loops

- **For loop** statement loops over each element in container one at a time, assigning a variable with next element that can then be used in the loop body
- **For loop** is used to iterating over sequence such as list, dictionary set or a string (iterables),
- **Example**: (print each items in the book list)

```
books = ["Scythe", "Harry Potter", "Narnia", "Hobbit"] # list
for x in books:                    # loop through list
    print(x)
```

```
Scythe
Harry Potter
Narnia
Hobbit
```

Review: For Loops

- **Loop through a string**
- **Example**:

```
str1 = ”hello world”
for x in str1:              # loop through a string
    print(x)
```

```
h
e
l
l
o

w
o
r
l
d
```

Review: For Loops (range function usage)

UNIVERSITY OF CALGARY

- **The range() function syntax**
- range(start, stop, step)
  - start value – optional
  - stop value – required
  - step value – optional

- **The range() function –** returns sequence of numbers starting from 0 by default and incremented by 1
- **Example 1**:

```
for x in range(5):          # here 5 is just stop value
    print(x)
```

```
0
1
2
3
4
```

Review: For Loops (range function usage)

UNIVERSITY OF CALGARY

- **Example 2**:

```
for x in range(3,5):        # here 3 is start and 5 is stop value
    print(x)
```

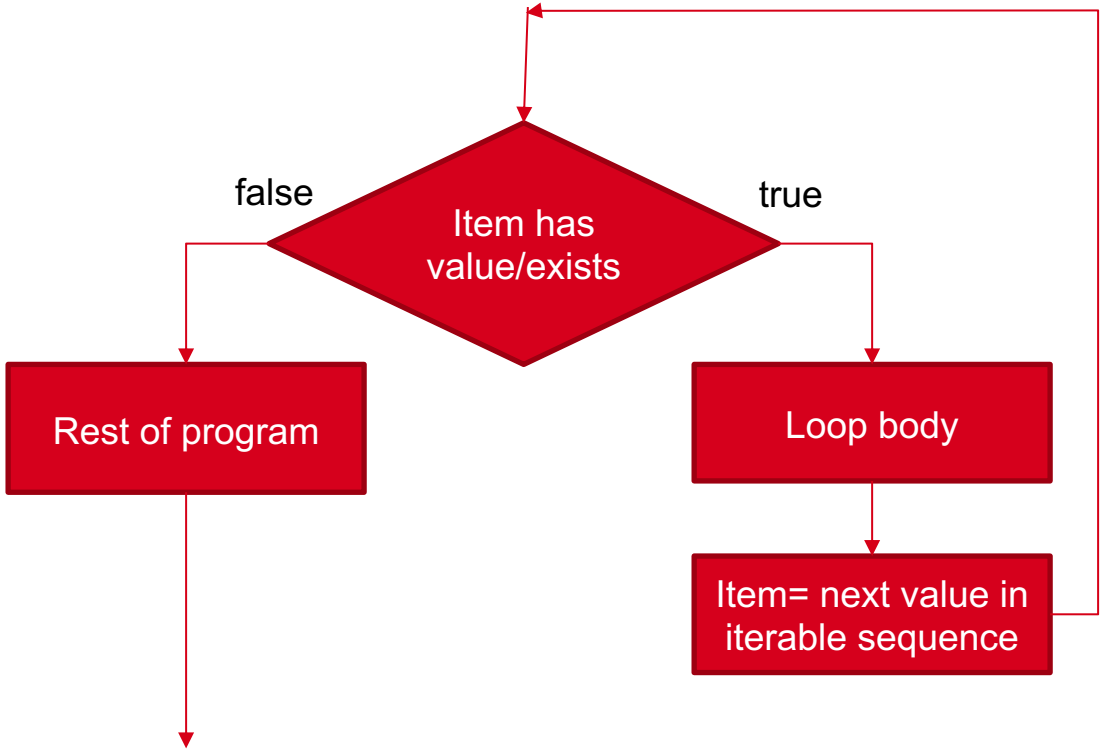
```
3
4
```

- **Example 3**:

```
for x in range(1,10, 2):    # here 1 is start, 10 is stop and 2 is step value
    print(x)
```

```
1
3
5
7
9
```

For loops in flowcharts



- Same break, continue, and else representation as while loops

Notes

- Why use range function?
  - A loop that uses multiple sequence structures
    - Same size
    - **Example**:

```
for x in range(5):              # here 5 is just stop value
    List3[x] = List1[x] + List2[x]
```

  - A loop that needs to repeat a process multiple times
    - Number of repetitions is not related to a sequence size
  - A loop that uses a systematically incremented variable
    - **Example**:

```
for x in range(1,n+1):          # to compute factorial n
    total*=x
```

Tutorial 6.1 – Password Modifier

UNIVERSITY OF CALGARY

- **Task** - Using a **While loop**, write a program that takes a simple password and makes it stronger by replacing characters using the key below, and by appending "!" to the end of the input string.
  - i becomes 1
  - a becomes @
  - m becomes M
  - B becomes 8
  - s becomes $

Tutorial 6.1 – Password Modifier

- Ex: if the input is:

  mypassword

- The output is:

  Myp@$$word!

Tutorial 6.1 – Password Modifier

UNIVERSITY OF CALGARY

Solution:

```
word = input()    #input the word
password = ""     # initialize the password
i = 0.            # initialize increment variable
```

Tutorial 6.1 – Password Modifier

UNIVERSITY OF CALGARY

Solution:

```
# while loop
while i < len(word):
    if word[i] == "i":
        password =
password + "1"
    elif word[i] == "a":
        password += "@"
    elif word[i] == "m":
        password += "M"
    elif word[i] == "B":
        password += "8"
```

Solution:

```
    elif word[i] == "s":
        password += "$"
    else:
        password += word[i]
    i = I + 1

password = password + "!"
print (password)
```

Tutorial 6.1 – Password Modifier

UNIVERSITY OF CALGARY

Solution using Dictionaries:

```
passkey = {‘i’ : ‘1’, ‘a’: ‘@’, ‘m’:’M’, ‘B’: ‘8’, ‘s’:’$’}    # dictionary keys:values
word = input()    #input the word.
 password = ""   # initialize the password
 i = 0.                # initialize increment variable
while i < len(word):          #while loop
            if word[ i ] in passkey.keys():
                        password += passkey[word[i]]
            else:
                        password += word[i]
            i = i+1
password += ‘!’
print(password)
```

Tutorial 6.2 – Output values in a list below a user defined amount

UNIVERSITY OF CALGARY

- **Task** - Write a program that first gets a list of integers from input. The input begins with an integer indicating the number of integers that follow.
- Then, get the last value from the input, which indicates a threshold.
- **Output** all integers less than or equal to that **last threshold value.**

Tutorial 6.2 – Output values in a list below a user defined amount

- **Ex:** if the input is:

```
5       # number of integers
50      # value1
60      # value 2
140     # value 3
200     # value 4
75      # value 5
100     # threshold
```

- The output is:

```
50, 60, 75,
```

Tutorial 6.2 – Output values in a list below a user defined amount

UNIVERSITY OF CALGARY

- Solution:

```
user_values = []
# Input begins with an integer indicating the number of integers that follow
num_values = int(input())
# Get list of integers from input
for i in range(num_values):
    user_input = int(input())
    user_values.append(user_input)
# Last value from the input indicates threshold
upper_threshold = int(input())
# Output all integers less than or equal to the threshold
for j in range(num_values):
    if user_values[j] <= upper_threshold:
        print(user_values[j], end=",")
```

Tutorial 6.3 – Print string in reverse

- **Task** – Write a program that takes in a line of text as input, and outputs that line of text in reverse.
- The program repeats, ending when the user enters "Done", "done", or "d" for the line of text.
- Ex: if the input is:

  Hello there
  Hey
  Done

- The output is:

  ereht olleH
  yeH

Tutorial 6.3 – Print string in reverse

UNIVERSITY OF CALGARY

```
# user input
user_input = input()
# loop through statements until user enter Done, done or d
while user_input != 'Done' and user_input != 'done' and user_input != 'd':
    output = ‘ ‘
      for i in range(len(user_input)-1,-1,-1):
            output = output + user_input[i]
      print(output)
      user_input = input()
```

Portfolio Assignment 2

- Due date: Oct 29th,2021 @ 11:59pm