Programming cont’d

IC152 Lecture 5
Feb 2021

Pseudo-code

1. Let n, sum, x be integers
2. Read n
3. Sum ← 0
4. Repeat n times
   - 4.1 Read x
   - 4.2 Sum ← sum+x
5. Print n, sum

Python code

```
n = input("Enter n: ")
sum = 0
for i in range(int(n)):
    x = input("Enter x: ")
    sum = sum + int(x)

print("n = ", n, "
   sum = ", sum)
```

What is this for?

How do the variables change?

Slide from IC152, 2018

n	sum	i	x
?	?	?	?
4	?	?	?
4	0	?	?
4	0	0	?
4	0	0	6
4	6	0	6
4	13	1	7
4	14	2	1
4	18	3	4
4	18	3	4

start

```
n = input()
sum = 0

for i in range(n):
    x = input()

    sum = sum + int(x)
```

```
for ...

for ...

for ...
```

Three operations happen here

The values of the variables before the program exits

`print` does not alter the memory

How to solve it by computer

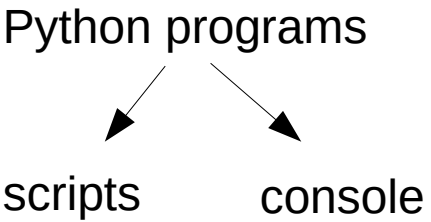
1. Understand the problem
2. Sketch out the algorithm
3. Write pseudocode
4. Write Python code
5. Test with various inputs to check

   Debugging: to fix errors in the program

Types of errors

- Syntax errors
  - eg. print(3+(2*9)
- Logical errors
  - May be more difficult to fix, but it depends
  - Will come with experience!

Functions



ipython console

```
In [48]: x = 32

In [49]: y = 'IIT'

In [50]: z = 4.17

In [51]: type(x)
Out[51]: int

In [52]: zi = int(z)

In [53]: zi
Out[53]: 4

In [54]: xf = float(x)

In [55]: xf
Out[55]: 32.0
```

Calling type() function with x as argument

Function to change the type of x

Math functions

```
In [57]: import math

In [58]: x = math.log10(1000)
```

```
In [60]: deg=45

In [61]: angle=deg*2*math.pi/360

In [62]: math.sin(angle)
Out[62]: 0.7071067811865475
```

User-defined functions

```
 8
 9  def newLine():
10      print()
11
12
13  print('First line')
14  newLine()
15  print('Second line')
```

function

main program

```
First line

Second line
```

```
 9  def newLine():
10      print()
11
12  def threeLines():
13      newLine()
14      newLine()
15      newLine()
16
17
18  print('First line')
19  threeLines()
20  print('Second line')
```

```
First line



Second line
```

Functions can be nested

```
 9  def newLine():
10      print('*')
11
12  def anyLines(n):
13      for i in range(n):
14          newLine()
15
16
17  print('First line')
18  anyLines(4)
19  print('Second line')
```

Functions can accept parameters (also called arguments)

Block 2

Block 1

```
First line
*
*
*
*
Second line
```