python™

Lecture (2)

Dr. Osama El-Ghonimy

Introduction to Programming

- What is programming?
- What is programming language?

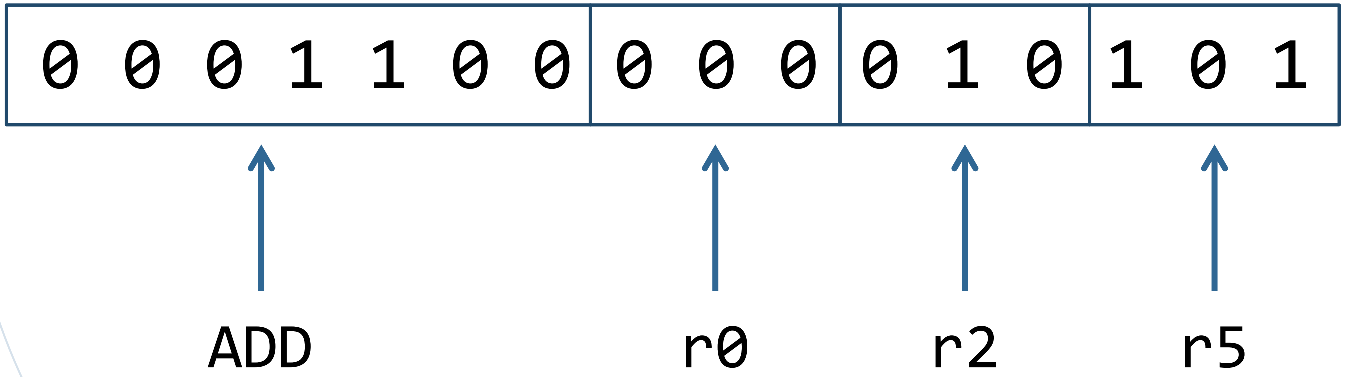
- The computer is binary. Numbers are written in binary. Letters are written as ASCII (binary). How to represent the commands?

Low Level Programming

- How to tell the computer to perform this addition?

```
r5 = r0 + r2
```

- The computer knows only binary. We need to convert this command into binary:



0 0 0 1 1 0 0	0 0 0	0 1 0	1 0 1
ADD	r0	r2	r5

- Changing any bit changes the meaning.

Low Level Programming

- Example of a program written in machine code:

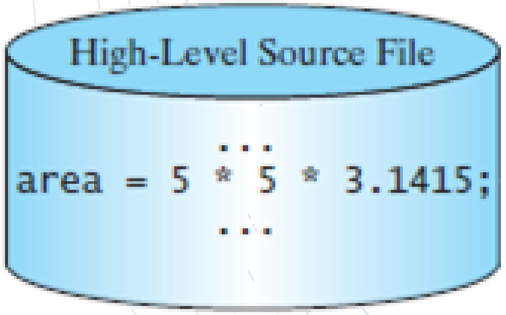
```
01100000000000000000000010000000
10100100000000000000000000000000
01100000000000001000000001000100
10100100000000010000000100000000
01100000000000100000000000000000
01100000000000110000000000000100
…
```

- This is machine code: low level programming.

High Level Programming

- You just type:

a = b + c



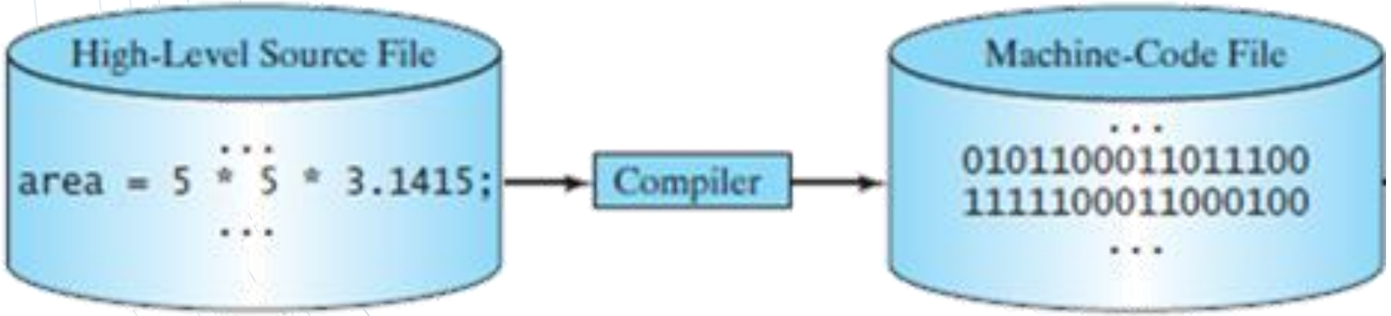
```
...
area = 5 * 5 * 3.1415;
...
```

High Level Programming

- You just type:

```
a = b + c
```

- The compiler convert this command into machine code sequence.

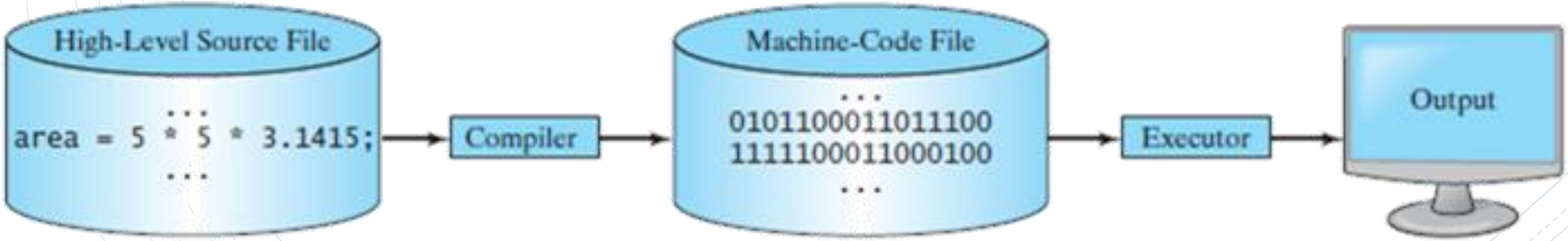


High Level Programming

- You just type:

a = b + c

- The compiler convert this command into machine code sequence.
- A program written in a high-level language is called a source program.



Setup

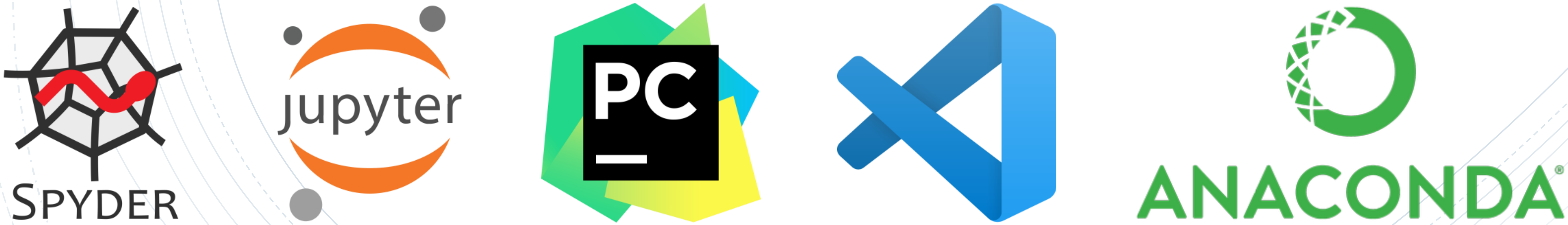
Part (A)

Python IDE

- Integrated Development Environment (IDE).
- Linux and Mac OS X users probably already have a usable Python preinstalled on their computers
- Spyder: close to MATLAB GUI.
- Jupyter: Web interface.
- PyCharm.

Python IDE

- Visual Studio Code.
- Another very popular Python distribution, particularly for math, science, engineering, and data science applications, is the Anaconda distribution.
- Anaconda includes all the most popular packages for engineering and data science type workloads in one single installer.



Setup Anaconda

- Download Anaconda:
  - Go to link: https://www.anaconda.com/products/distribution .
  - Download for windows x64.
  - Double click the downloaded file >> Next >> I agree >> Next >> Next >> Select “Add Anaconda3 to my PATH environment variable” >> Install >> Finish.
- Open Anaconda: Start menu >> Anaconda3 (64-bit) >> Anaconda Navigator (anaconda3).

Setup Visual Studio Code

- Download VS Code:
  - Go to link: https://code.visualstudio.com/download .
  - Download for windows x64.
  - Double click the downloaded file >> Next >> Next >> Next >> Install >> Finish.
- Open VS Code:
  - Start menu >> Visual Studio Code.

Setup Visual Studio Code

- Install Python extension:
  - Type “Python” in the search box.
  - Select the first search result (Python), Developer: Microsoft.
  - Click on install.
- Open New File:
  - File menu >> New File >> Jupyter Notebook.
  - If VS code recommended installing any other extension, accept and install.

Arithmetic Operators

Part (B)

Arithmetic Operators

- Addition:

$$5 + 3$$

- Subtraction:

$$5 - 3$$

- Multiplication:

$$5 \times 3$$

```
>>> 5+3
8

>>> 5-3
2

>>> 5*3
15
```

Arithmetic Operators

- Division:

$$5 \div 3 = 1\frac{2}{3}$$

  - Normal division (True division).
  - Integer division (Floor division).

- Reminder (modulus):

$$6 \div 3 = 2\frac{0}{3}$$

```
>>> 5/3
1.6666666666666667

>>> 5//3
1

>>> 5%3
2

>>> 6%3
0
```

Arithmetic Operators

- Reminder (modulus):

$$6 \div 4 = 1\frac{1}{2} = 1\frac{2}{4}$$

- Find a quotient and remainder simultaneously.

- Power:

$$5^3$$

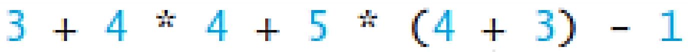
```
>>> 6%4
2
```

```
>>> divmod(5,3)
(1, 2)
```

```
>>> 5**3
125
```

Order of Precedence

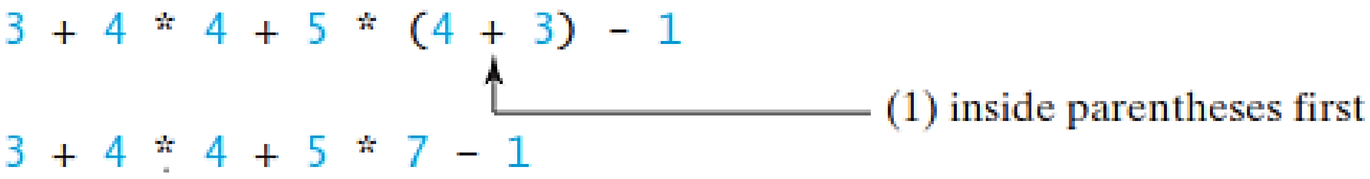
- What is the output of this expression evaluation?



```
3 + 4 * 4 + 5 * (4 + 3) - 1
```

ORDER OF PRECEDENCE

- What is the output of this expression evaluation?
  - Operators contained within pairs of parentheses are evaluated first.



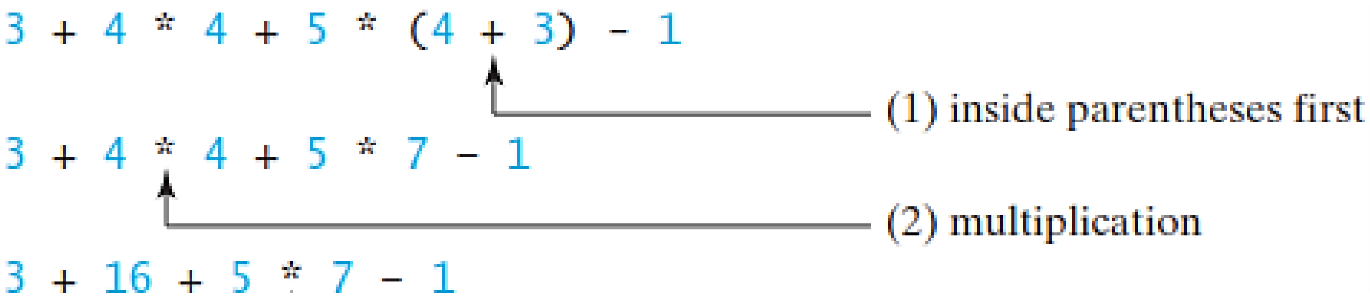
```
3 + 4 * 4 + 5 * (4 + 3) - 1
```

(1) inside parentheses first

```
3 + 4 * 4 + 5 * 7 - 1
```

ORDER OF PRECEDENCE

- What is the output of this expression evaluation?
  - Multiplication, division, and remainder operators are applied next.



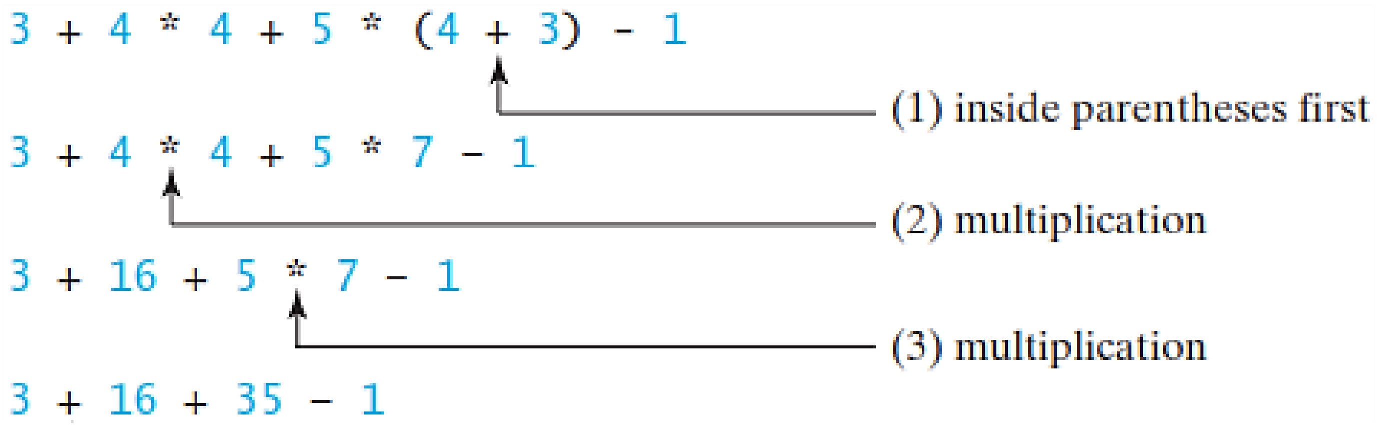
```
3 + 4 * 4 + 5 * (4 + 3) - 1

3 + 4 * 4 + 5 * 7 - 1

3 + 16 + 5 * 7 - 1
```

(1) inside parentheses first

(2) multiplication

ORDER OF PRECEDENCE

- What is the output of this expression evaluation?
  - Several multiplication, division, and remainder are applied in order.



```
3 + 4 * 4 + 5 * (4 + 3) - 1

3 + 4 * 4 + 5 * 7 - 1

3 + 16 + 5 * 7 - 1

3 + 16 + 35 - 1
```

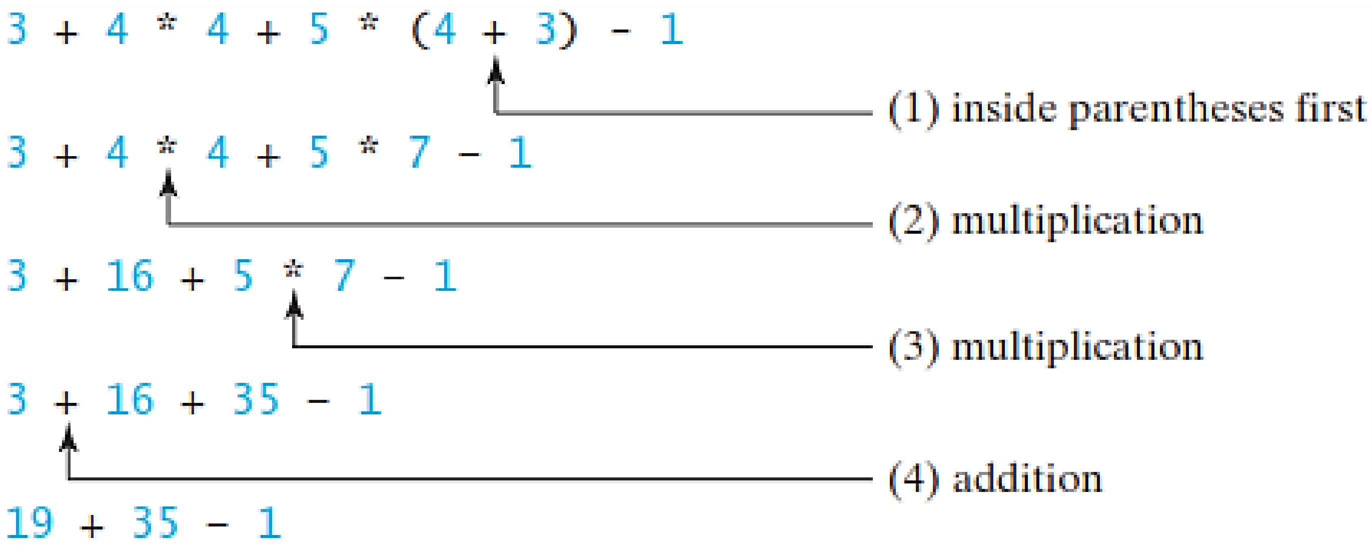
(1) inside parentheses first

(2) multiplication

(3) multiplication

Order of Precedence

- What is the output of this expression evaluation?
  - Addition and subtraction operators are applied last.



```
3 + 4 * 4 + 5 * (4 + 3) - 1
3 + 4 * 4 + 5 * 7 - 1
3 + 16 + 5 * 7 - 1
3 + 16 + 35 - 1
19 + 35 - 1
```

(1) inside parentheses first

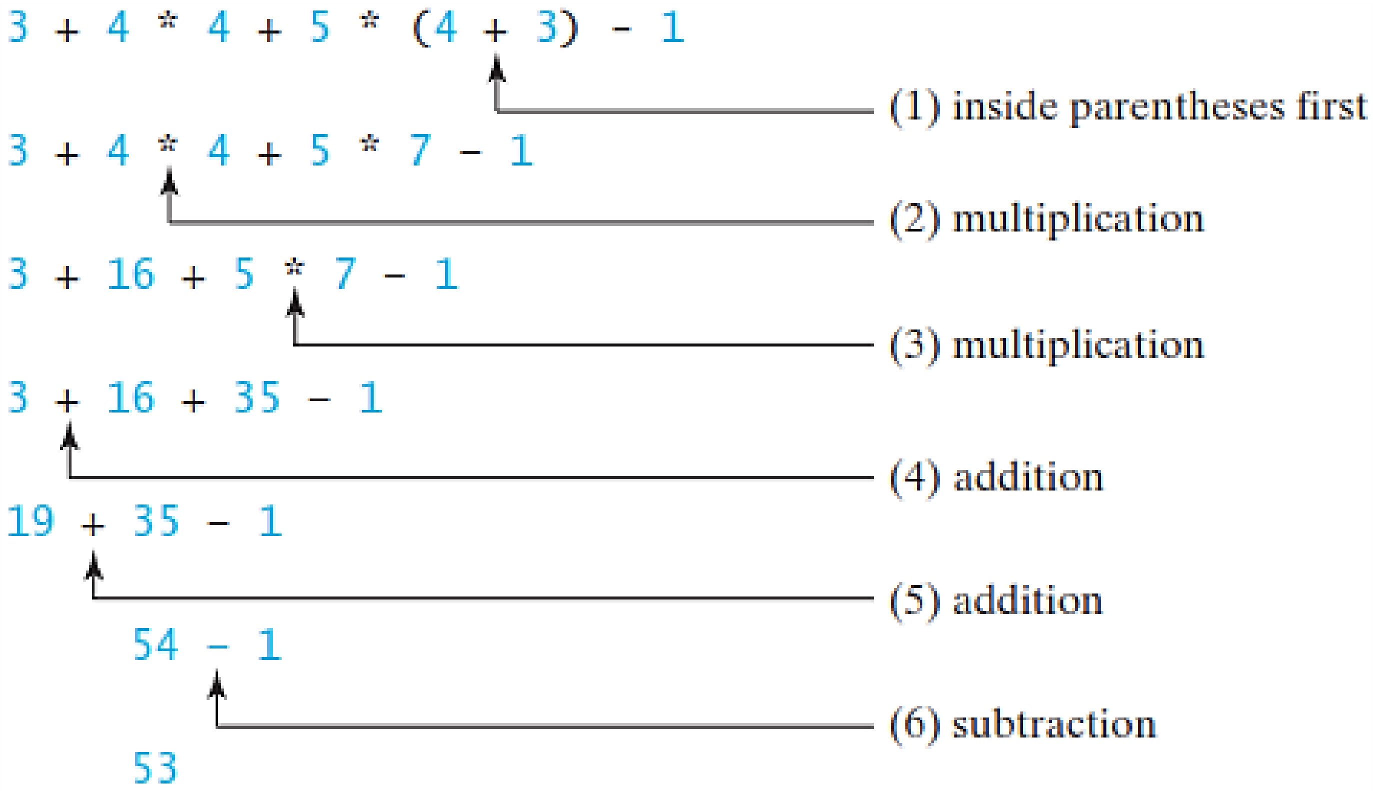
(2) multiplication

(3) multiplication

(4) addition

Order of Precedence

- What is the output of this expression evaluation?
  - Several additions and subtractions are applied from right to left.



```
3 + 4 * 4 + 5 * (4 + 3) - 1
3 + 4 * 4 + 5 * 7 - 1
3 + 16 + 5 * 7 - 1
3 + 16 + 35 - 1
19 + 35 - 1
    54 - 1
    53
```

(1) inside parentheses first

(2) multiplication

(3) multiplication

(4) addition

(5) addition

(6) subtraction

Cells

- Only the last output is displayed.
- How to solve this?

```
10+4
10-6
```

```
4
```

- Solution (1): Create multiple cells.

```
10+4
```

```
14
```

```
10-6
```

```
4
```

Print

- Only the last output is displayed.
- How to solve this?

```
10+4
10-6
```

```
4
```

- Solution (2): Print all but the last output.

```
print(10+4)
10-6
```

```
14
4
```

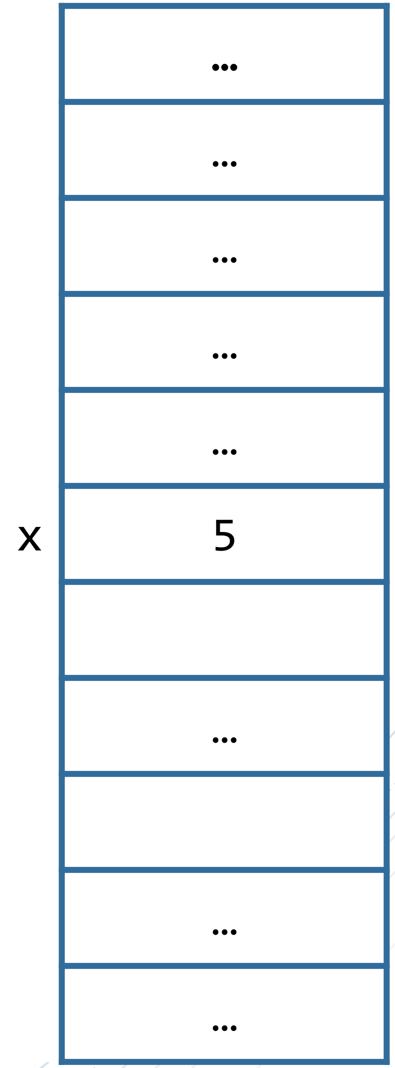
Variables, Assignment, and Comments

Part (C)

Creating Variables

- Any application consists of code and data.
- Create variable from:
  - Instant value.
- The variable is stored in computer memory (RAM):
  - The variable name is a reference for the data in memory.

```
>>>  x=5
```

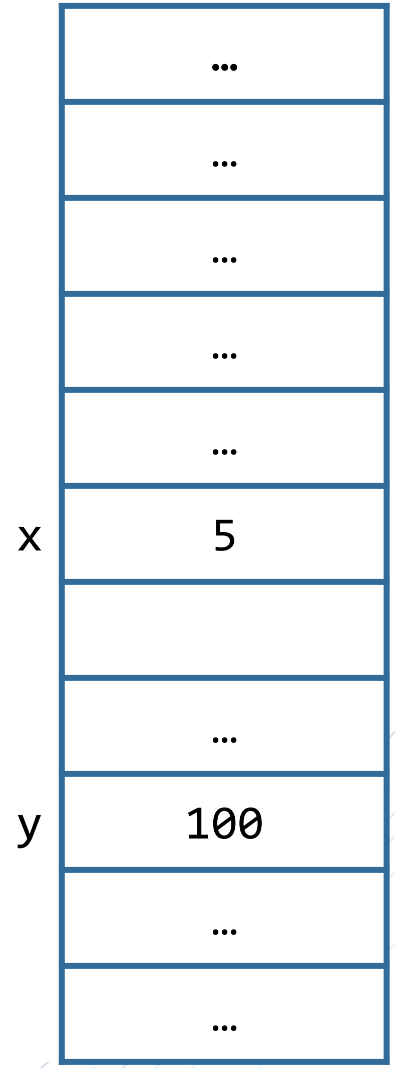


	…
	…
	…
	…
	…
x	5
	…
	…
	…

Creating Variables

- Any application consists of code and data.
- Create variable from:
  - Instant value.
  - An expression.
- The variable is stored in computer memory (RAM):
  - The variable name is a reference for the data in memory.

```
>>> x=5

>>> y=5*20
```

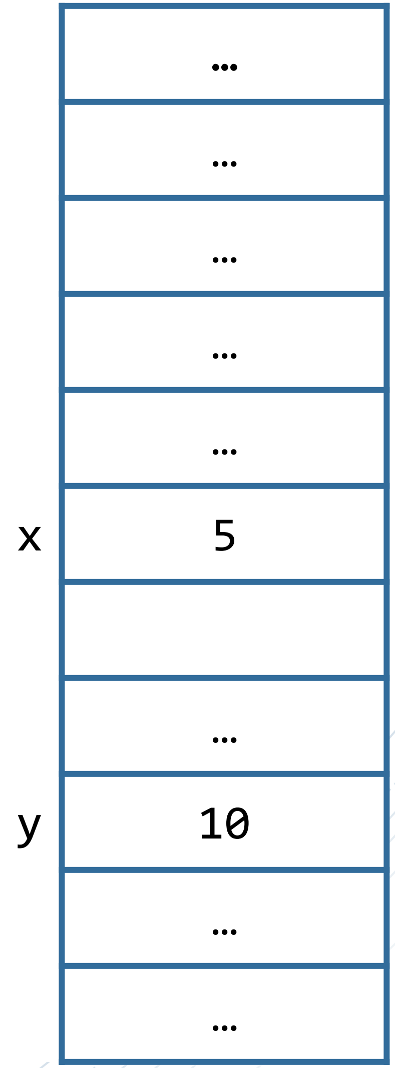


	…
	…
	…
	…
	…
x	5
	…
y	100
	…
	…

Creating Variables

- Create variable from:
  - The expression may include another variable (or more).
  - Note that the new value overwrites the old one.

```
>>> x=5

>>> y=5*20

>>> y=5+x
```

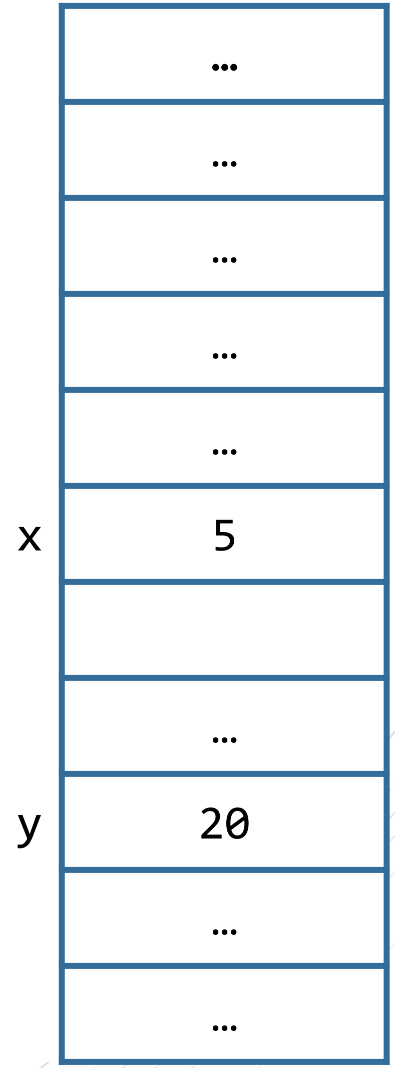


	…
	…
	…
	…
	…
x	5
	…
y	10
	…
	…

Creating Variables

- Create variable from:
  - The expression may include another variable (or more).
  - Note that the new value overwrites the old one.
  - The expression may include the same variable (old value).

```
>>> x=5

>>> y=5*20

>>> y=5+x

>>> y=y*2
```



	…
	…
	…
	…
	…
x	5
	…
y	20
	…
	…

Creating Variables

- Create variable from:
  - The expression may include another variable (or more).
  - Note that the new value overwrites the old one.
  - The expression may include the same variable (old value).
- Note that there is no implicit multiplication.

```
>>> x=5

>>> y=5*20

>>> y=5+x

>>> y=y*2

>>> y=2y
       ^ SyntaxError: invalid
syntax
```

Example

- Write a program to calculate the area and circumference of a circle based on the following equations:

$$area = \pi r^2$$

$$circumference = 2\pi r$$

- Where:

$$\pi = 3.14159, \qquad r = 10m$$

- Only the last output is displayed. Solution??

```
r = 10

3.14159*r*r

2*3.14159*r
```

Output:

```
62.8318
```

Example

- Write a program to calculate the area and circumference of a circle based on the following equations:

$$area = \pi r^2$$

$$circumference = 2\pi r$$

- Where:

$$\pi = 3.14159, \qquad r = 10m$$

- The user does not know which output is area or circumference.

```
r = 10

print(3.14159*r*r)

2*3.14159*r
```

Output:

```
314.159
62.8318
```

Example

- Write a program to calculate the area and circumference of a circle based on the following equations:

$$area = \pi r^2$$

$$circumference = 2\pi r$$

- Where:

$$\pi = 3.14159, \qquad r = 10m$$

- The user does not know which output is area or circumference.

```
r = 10
print("Area =")
print(3.14159*r*r)

2*3.14159*r
```

Output:

```
Area =
314.159
62.8318
```

Example

- Write a program to calculate the area and circumference of a circle based on the following equations:

$$area = \pi r^2$$

$$circumference = 2\pi r$$

- Where:

$$\pi = 3.14159, \qquad r = 10m$$

- Print the two values on the same line using single print statement.

```
r = 10
print("Area =", 3.14159*r*r)


2*3.14159*r
```

Output:

```
Area = 314.159
62.8318
```

Example

- Write a program to calculate the area and circumference of a circle based on the following equations:

$$area = \pi r^2$$

$$circumference = 2\pi r$$

- Where:

$$\pi = 3.14159, \qquad r = 10m$$

- Same for circumference.
- How to add units?

```
r = 10
print("Area =", 3.14159*r*r)

print("Circumference =",
2*3.14159*r)
```

Output:

```
Area = 314.159
Circumference = 62.8318
```

Example

- Write a program to calculate the area and circumference of a circle based on the following equations:

$$area = \pi r^2$$

$$circumference = 2\pi r$$

- Where:

$$\pi = 3.14159, \qquad r = 10m$$

- Same for circumference.
- How to add units?

```
r = 10
print("Area =", 3.14159*r*r,
"m2")
print("Circumference =",
2*3.14159*r, "m")
```

Output:

```
Area = 314.159 m2
Circumference = 62.8318 m
```

Example

- Write a program to calculate the area and circumference of a circle based on the following equations:

$$area = \pi r^2$$

$$circumference = 2\pi r$$

- Where:

$$\pi = 3.14159, \qquad r = 10m$$

- One last note: using the **power operator**.

```
r = 10
print("Area =", 3.14159*r**2, "m2")
print("Circumference =", 2*3.14159*r, "m")
```

Output:

```
Area = 314.159 m2
Circumference = 62.8318 m
```

Comments

- Provide some information about code.
- Syntax: Any # inside the code starts a comment till the end of line.
- The compiler skips compiling these lines. No syntax error.
  - Without # it is a command then there will be an error.
- It is also used for debugging.

```
# A program to find area and
# circumference of a circle
# By Dr. Osama El-Ghonimy

r = 10      # radius
print("Area =", 3.14159*r**2,
"m2")
print("Circumference =",
2*3.14159*r, "m")
```

Output:

```
Area = 314.159 m2
Circumference = 62.8318 m
```