Technical University of Lodz
Institute of Electronics

Algorithms and Data Structures

3. Program Components and Data Types

Łódź 2012

Exercise – Body Mass Index

- Type in the program
- Save it as **bmi.py**
- Run the script

```
1  # BMI.py
2
3  YourName = input( "Enter your name: " )
4  var = input ( "Enter your height [in cm]: " )
5  YourHeight = var / 100.0
6  YourMass = input ( "Enter the mass of your body [in kg]: " )
7  BMI = YourMass / ( YourHeight * YourHeight)
8  print ("\n")
9  print ("Hello " + YourName + "!")
10 print "Your Body Mass Index is %4.1f" % ( BMI )
11
```

>>> Enter your name: ”John”

>>> Enter your height [in cm]: 185

>>> Enter the mass of your body [in kg]: 87

Variables

- ***Variables*** are names that refer to data stored in the memory.
- Variables are also named ***identifiers*** in Python.

```
1  # BMI.py
2
3  YourName = input( "Enter your name: " )
4  var = input ( "Enter your height [in cm]: " )
5  YourHeight = var / 100.0
6  YourMass = input ( "Enter the mass of your body [in kg]: " )
7  BMI = YourMass / ( YourHeight * YourHeight)
8  print ("\n")
9  print ("Hello " + YourName + "!")
10 print "Your Body Mass Index is %4.1f" % ( BMI )
11
```

- There are 5 variables used in program *BMI.py*:

YourName, var, YourHeight, YourMass, BMI

Variable names

Variable names

- may consist of lower or upper alphabetic characters, the underscore _, and the digits 0-9,
- can not start with a digit,
- can not be Python ***keywords***

and	**del**	**from**	**not**	**while**
as	**elif**	**global**	**or**	**with**
assert	**else**	**if**	**pass**	**yield**
break	**except**	**import**	**print**	
class	**exec**	**in**	**raise**	
continue	**finally**	**is**	**return**	
def	**for**	**lambda**	**try**	

To make the program easier to follow, define meaningful identifiers.

Program statements

- Program is a sequence of ***statements***.
- Two types of statements are used in *bmi.py* program:
  assignment statement and **print()** statement.

Assignment statements: <variable> = <expression>

Read assignment statement from right to left:

1) Evaluate the expression on the right.
2) Assign the variable on the left to refer to that value.

```
>>> #Computes the right side to be -2 and assigns x to refer to -2
>>> x = 10 – 17 + 5
```

An assignment statement ”=” is not like a math „equals” sign.
It is rather a delimiter to separate the variable from expression.

Program statements

Print statements: print <expression1>, <expression2>, …

are used to produce program output.

```
>>> #Expression in quotation marks (so-called strings) are printed literally.
>>> print ”Ala ma kota.”
>>> print ’Ala ma kota.’

>>> #Expressions without quotation marks have their value printed.
>>> n = 5
>>> print ‘Ala ma kota oraz‘, n-3, ‘ psy.’
```

- Expressions are separated by single spaces.
- Each print statement produces output in a separate line.

Program statements

- Type in and run the following script

```
1  # tmp.py
2
3  from math import pi
4  r = 12
5  area = pi * r ** 2
6  print "The area of a circle with radius", r, "is", area
7
```

- Change the **print()** statement to

  >>> print ”The area of a circle with radius r is area”

  Explain the result.

- Technically, **print()** is a built-in function that is called to incorporate a print statement into a program.

Data types

- A variable in Python may refer to any type of data. Three examples of data types are **strings**, **integers** and **floats**.

Strings are sequences of characters inside single or double quotation marks.

```
>>> Motto = ”Programming is fun!”
>>> aQuestion = ’Does she like studying at IFE?’
```

Integers are whole number values.

```
>>> Large_Score = 857
>>> NegativeResult = -39
```

Floats (floating point) are numbers with a fractional part.

```
>>> DistanceInMeters = 857.21
>>> SmallQuantity = -0.000138
```

Expressions

Input expression **input(prompt)**

is used to produce program intput. It displays **prompt** and returns user input as a string. The **prompt** is printed to alert the user that the program is waiting for input.

```
>>> age = input (”Enter your age: ”)
>>> print ”When I was”, age, ”strange thing happened!)”
```

Expressions

Numeric expressions use the arithmetic operations **+, -, *, /, //, **, %** ,

respectively for addition, subtraction, multiplication, division, integer division, raising to a power, and modulo (remainder from division of integer numbers).
Rules of precedence are:

1) exponentiation,
2) multiplication, division and modulo (left to right),
3) addition and subtraction (left to right).

Parentheses may be used to change the order of operations.

- Integer division rounds down to the nearest integer

```
>>> 7//2
>>> -1//3
```

- Integer division of floats rounds down to an integer, but result is of float type.
- Variables must be assigned a value before being used in an expression.

Comments

Comments begin with a number (hash) sign # and signify that whatever follows is to be ignored by the inpterpreter.

```
# Text that helps explain your program to others
```

Comments can be placed everywhere in a Python program. They explain the code for a human reader rather than for the interpreter.

The Python **interpreter** translates the program code to the form that can be executed by the computer. When one starts the Python shell or PyLab, the interpreter displays a prompt sign (e.g. **>>>**). This means it is ready to translate any input entered after the promt.

>>> print „Hi, buddy!”

Exercises

3.1. Give 3 names that are legal and 3 names that are not legal Python identifiers.

3.2. List each variable in the program below and give the type of data that is refered to. Identify the expressions and assignment statements

```
1  # tmp.py
2
3  from math import pi
4  r = 12
5  area = pi * r ** 2
6  print "The area of a circle with radius", r, "is", area
7
```

3.3. Explains what happens if the final space is deleted from the input statement

>>> k = input (”Enter k: ”)

Exercises

3.4. Determine the value of each of these Python expressions

(a) **1 + 2 * 3 – 4 * 5 + 6**

(b) **1 + 2 ** 3 * 4 - 5**

(c) **1 / 2 – 3 / 4**

(d) **1 // 2 – 3 //4 + 5 // 6**

(e) **1 + 4 – 2 / 2**

(f) **(1 + 4 – 2) / 2**

3.5. Determine the output of each of these code fragments

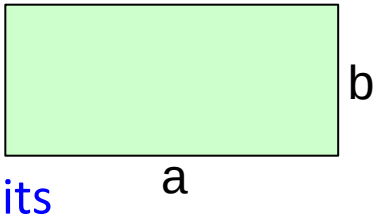
(a)
```
x = 10
y = 15
print x, y
y = x
print x, y
x = 5
print x, y
```

(b)
```
x = 10
y = 15
print x, y
y = x
x = y
print x, y
x = 5
print x, y
```

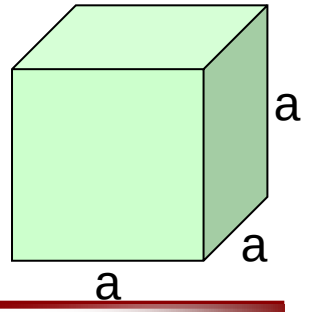
(c)
```
x = 10
y = 15
z = 20
x = z
z = y
y = x
print x, y, z
```

Exercises

3.6. Write a program **average.py** that calculates and prints the average of 3 integer numbers. Print the result as a floating-point number with two digits after the decimal separator. Modify the number of displayed fraction digits.
Hint: Use a **print** statement similar to the one used in line 10 of **BMI.py** program.

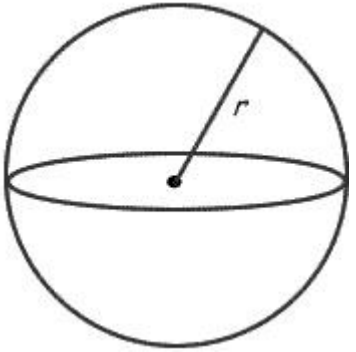


3.7. Modify listing **average.py** to write a program **rect.py** that calculates and prints the area and perimeter of a rectangle given its length and width. Run your program with several different values of the length and width to test it.



3.8. Modify listing **average.py** to write a program **cube.py** that calculates and prints the volume and surface area of a cube given its width. Run your program with several different values of the width to find a point at which volume equals (numerically) surface area.

Exercises



3.9. Modify listing **average.py** to write a program **sphere.py** that calculates and prints the volume and surface area of a sphere given its radius. Run your program with different values of the radius to find the point at which volume equals (numerically) surface area.

3.10. Write a Python script to print a box like the one below.

```
**********
*        *
**********
```

3.11. Having a „tip calculator” in your mobile phone would be useful Write one assuming there is a Python interpreter available on the phone. Ask the user for the price of the meal and the percent tip they want to leave. Then print both the tip amount and the total bill with the tip included.

Summary

1) Variables refer to …
2) … via assigment statements. Remember to read them from right to left.
3) Data can be of type **string**, **integer** or **float** (so far). The data type of a variable determines what one can do with it.
4) **input()** is an expression to enter data into a running program (in a Python console mode).
5) Arithmetic operations are used in numeric expressions. They follow rules of precedence. One can change the order of operations in an expression by using parentheses.
6) The ”=” symbol denotes assignment, not mathematical „equal” sign.
7) Comments are ignored by the Python interpreter. A program script should be rich in comments. (Why?)

Literature

Brian Heinold, Introduction to Programming Using Python, Mount St. Mary's University, 2012 (*http://faculty.msmary.edu/heinold/python.html*).

Brad Dayley, Python Phrasebook: Essential Code and Commands, SAMS Publishing, 2007 (dostępne też tłumaczenie: B. Dayley, Python. Rozmówki, Helion, 2007).

Mark J. Johnson, A Concise Introduction to Programming in Python, CRC Press, 2012.