Assign the value 5 to a, and value 6 to b. Assign the value of a + b to variable c.

**Examples**

>>> a

5

>>> b

6

>>> c

11

a =5

b =6

c =a+b

Floating point number is represented with a dot(.) followed by one or more decimals (can be zero).

**Examples**

**>>>** 3/2

1

**>>>** 3.0/2

1.5

**>>>** pi

3.14

**>>>** type(pi)

< type 'float'>

**>>>** area

28.26

**>>>** perimeter

18.84

#Compute the area and perimeter of a circle with radius = 3

pi = 3.14

area =pi\*3\*\*2

perimeter =2\*pi\*3

Numbers that contains decimal point are called *floating point* numbers. The type(x) function will return if the argument x is a float. You can use the float(x) and int(x) function to convert values between float and integer.

**Examples**

**>>>** type(5)

<type 'int'>

**>>>** float(5)

5.0

**>>>** type(float(5))

<type 'float'>

# Change the type of the variable x to float

# Change the type of variable y to integer

x = 123446754336788543835697

y = 3.14159265358979323846

x =float(x)

y =int(y)

Declare a string literal by enclosing the literal using single, double or triple quotes. Triple quote allows the literal to span multiple lines.

**Examples**

>>> spam = 'mail'

>>> contents = """This is Line 1

... and this is Line 2

... and this is Line 3

... """

>>> spam

'mail'

>>> contents

'This is Line 1\nand this is Line 2\nand this is Line 3\n'

>>> 'How're you?' *# Syntax error for same quote*

**SyntaxError**: invalid syntax

>>> "How're you?" *# Using different quote*

"How're you?"

>>> 'How**\'**re you?' *# Escapte quote with \*

"How're you?"

>>> foobar

'"No, thanks, Mom," I said, "I don**'**t know how long it will take."'

# Assign foobar which gives the output shown in the last example.

# Hint: Use the triple quote as the outermost quote

foobar =

Topic 1: Question 6

Certain mathematical operations such as addition and multiplication can be used on the String datatype. Study the examples given below to see how it works:

**Examples**

>>> 5 + 6 *# Adding numbers*

11

>>> '5' + '6' *# Adding strings*

'56'

>>> 5 \* 6 *# Multiplying numbers*

30

>>> '5' \* 6 *# Multiplying string with number*

'555555'

# Assign 'HelloWorld!' to variable a

a ='HelloWorld!'

# b contains 'HelloWorld!HelloWorld!HelloWorld!HelloWorld!HelloWorld!'

b = a\*5

Topic 1: Question 7

You can make use of the len(x) function to find out the number of characters in a string.

**Examples**

>>> greetings = "Hello World"

>>> len(greetings) *# get the length of string*

11

>>> greetings[0] *# get the 1st character*

'H'

greeting = "Hello Google!"

# number of characters stored in the variable greeting

number\_of\_char = len(greeting)

# repeat the greetings based on the number of character in 'greeting'

greetings = number\_of\_char\*greeting

Topic 1: Question 8

Adding two strings or making multiple copies of the same string.

**Examples**

>>> greetings = "Hello World"

>>> len(greetings) *# get the length of string*

11

>>> greetings[0] *# get the 1st character*

'H'

>>> "@@" + "@@@" *# add strings or characters using '+' operator*

'@@@@@'

>>> "@" \* 5 *# make mutliple copies using '\*' operator*

'@@@@@'

>>> **print** underline("Good Day")

Good Day

\_\_\_\_\_\_\_\_

>>>

# Write a function, given a string of characters, return the string together with '\_'s of the same length.

def underline(title):

return title+'\n'+len(title)\*'\_'

Topic 1: Question 9

Introducing some string methods.

**Examples**

>>> greetings = "Hello World"

>>> greetings.lower() *# convert to lower case*

'hello world'

>>> greetings.find('o') *# return first occurence of character or substring*

4

>>> greetings.replace('World', 'Everyone')

'Hello Everyone'

>>> spam = ' This sentence has leading and trailing spaces. \n'

>>> spam

' This sentence has leading and trailing spaces. \n'

>>> spam.strip()

'This sentence has leading and trailing spaces.'

# Use one or more string methods in above examples, extract the substring

# surrounded by 'xyz' at the beginning and end. Replace the ',' in the substring with '|'.

# and remove all trailing space.

str1 = 'abcefghxyzThis,is,the,target,string xyzlkdjf'

idx1 = str1.find( 'xyz' ) # get the position of 'xyz'

idx2 = str1.find('xyz', idx1+1) # get the next 'xyz'

str1 = str1[idx1+3:idx2].replace(',','|') # replace ',' with '|'

str1 = str1.strip() # strip trailing spaces.

Topic 1: Question 10

Like other programming languages, Python also has some basic types like numbers, strings, lists and dictionaries.

**Examples**

>>> type(a)

<type 'str'>

>>> type(b)

<type 'int'>

>>> type(c)

<type 'float'>

>>> type(d)

<type 'list'>

# Assign arbitrary values to the variables such that they are of the types used in the examples

a ='a'

b =2

c =2.0

d =['a','b','c']

Topic 1: Question 12

A integer or floating-point number with trailing 'j' or 'J' is a complex number.

**Examples**

>>> a = 1 + 1j

>>> type(a)

<type 'complex'>

# Compute the sum and product of 2 complex numbers:

# (2+3j) and (4+5j)

a = (2+3j)

b =(4+5j)

sum\_ab =a+b

prod\_ab =a\*b

Topic 1: Question 13

Format string output by using the '%' operator

**Examples**

>>> 1.0/3

0.33333333333333331

>>> **print** "**%.2f**" % (1.0/3) *# Convert to floating point with 2 decimal places*

0.33

>>> name = "John"

>>> age = 27

>>> **print** "My name is **%s**. I am **%d** years old." % (name, age)

My name **is** John. I am 27 years old.

>>> **print** "**%03d**" % 5 *# zero padded if output is less than 3 digits.*

005

>>> dec2hex(11)

'0x0b'

# Write a function that does a decimal to hexadecimal conversion.

# Hint: Make use of "%x" for hexadecimal format.

def dec2hex(num):

return('0x' + '%.2x' % num)

Topic 1: Question 14

Accessing string elements.

A string is a sequence of characters.

Each character can be retrieved using an integer index, starting from zero.

To access a substring use s[i:j], which returns a substring from index i to

index (j-1).

**Examples**

>>> greetings = "Hello world"

>>> len(greetings) *# get the length of string*

11

>>> greetings[0] *# get the 1st character*

'H'

>>> greetings[0:2] *# get first two character*

'He'

# Extract each word from 'greetings' and assign to

# variables 'first', 'middle' and 'last'.

greetings = "How are you"

first = greetings[0 :3 ]

middle = greetings[4 :7 ]

last = greetings[8:11 ]

Topic 1: Question 15

Octal and hexadecimal integer.

The default literal representation is in decimal format. To represent a octal or hexadecimal literal,

precede the value with '0' and '0x' respectively.

**Examples**

>>> *# Assign the value of 25 using decimal, octal and hexadecimal to a, b, and c respectively.*

>>> a

25

>>> b

25

>>> c

25

a = 25

b = 0

c = 0x

Topic 2: Question 1

Define a function calls addNumber(x, y) that takes in two number and returns the sum of the two numbers.

**Examples**

>>> addNumber(2, 4)

6

>>> addNumber(-2, -4)

-6

>>> addNumber(-2, 2)

0

def addNumber(x, y):

return x+y

Topic 2: Question 2

Define a function calls subtractNumber(x, y) that takes in two numbers and returns the difference of the two numbers.

**Examples**

>>> subtractNumber(20, 7)

13

>>> subtractNumber(-20, -4)

-16

>>> subtractNumber(-2, -2)

0

def subtractNumber(x, y):

return x-y

Topic 2: Question 3

Write a function getBiggerNumber(x, y) that takes in two numbers as arguments and returns the bigger number.

**Examples**

>>> getBiggerNumber(20, -10)

20

>>> getBiggerNumber(10, 10)

10

>>> getBiggerNumber(10, 20)

20

def getBiggerNumber(x,y):

return max(x,y)

Topic 2: Question 4

Python provides many built-in modules with many useful functions. One such module is the math module. The math module provides many useful functions such as sqrt(x), pow(x, y), ceil(x), floor(x) etc. You will need to do a "import math" before you are allowed to use the functions within the math module.

**Examples**

>>> **import** **math**

>>> math.sqrt(4)

2.0

>>> math.pow(2, 3)

8.0

>>> math.pi

3.1415926535897931

import math

# Calculate the square root of 16 and stores it in the variable a

a =math.sqrt(16)

# Calculate 3 to the power of 5 and stores it in the variable b

b =3\*\*5

# Calculate area of circle with radius = 3.0 by making use of the math.pi constant and store it in the variable c

c =math.pi\*pow(3,2)

Topic 2: Question 5

Write a function to convert temperature from Celsius to Fahrenheit scale.  
**oC to oF Conversion:** Multipy by 9, then divide by 5, then add 32.

**Examples**

>>> Cel2Fah(28.0)

'82.40'

>>> Cel2Fah(0.00)

'32.00'

# Note: Return a string of 2 decimal places.

def Cel2Fah(temp):

f=32+temp\*9/5

return format(f,'.2f')

Topic 2: Question 6

Write a function to compute the BMI of a person.  
    BMI = weight(kg)  /  ( height(m)\*height(m) )

**Examples**

>>> BMI(63, 1.7)

'21.8'

>>> BMI(110, 2)

'27.5'

# Note: Return a string of 1 decimal place.

def BMI(weight, height):

bmi = float(weight)/(height\*height)

return '%.1f' % bmi

Topic 2: Question 7

Write a function percent(value, total) that takes in two numbers as arguments, and returns the percentage value as an integer.

**Examples**

>>> percent(46, 90)

51

>>> percent(51, 51)

100

>>> percent(63, 12)

525

def percent(x,y):

a=100\*x/y

return a

Topic 2: Question 8

The Pythagoras' Theorem for a right-angle triangle can be written as a2+b2 = c2, where a and b are sides of the right angle and c is the hypotenuse. Write a function to compute the hypotenuse given sides a and b of the triangle.

**Examples**

>>> hypotenuse(3, 4)

5

>>> hypotenuse(5, 12)

13

# Hint: You can use math.sqrt(x) to compute the square root of x.

import math

def hypotenuse(a, b):

c=math.sqrt((a\*\*2)+(b\*\*2))

return c

Topic 2: Question 9

Write a function getSumOfLastDigits() that takes in a list of positive numbers and returns the sum of all the last digits in the list.

Topic 2: Question 9

Write a function getSumOfLastDigits() that takes in a list of positive numbers and returns the sum of all the last digits in the list.

**Examples**

>>> getSumOfLastDigits([2, 3, 4])

9

>>> getSumOfLastDigits([1, 23, 456])

10

def getSumOfLastDigits(numList):

Topic 2: Question 10

Write a function that uses a default value.

**Examples**

>>> introduce('Lim', 20)

'My name is Lim. I am 20 years old.'

>>> introduce('Ahmad')

'My name is Ahmad. My age is secret.'

def introduce(name, age=0):

msg = "My name is %s. " % name

if age == 0:

msg += "My age is secret."

else:

msg += "I am %d years old." % age

return msg

Topic 2: Question 11

Write a function isEquilateral(x, y, z) that accepts the 3 sides of a triangle as arguments. The program should return True if it is an equilateral triangle.

**Examples**

>>> isEquilateral(2, 4, 3)

False

>>> isEquilateral(3, 3, 3)

True

>>> isEquilateral(-3, -3, -3)

False

def isEquilateral(x, y, z):

if x==y==z>0:

return True

else:

return False

Topic 2: Question 12

For a quadratic equation in the form of ax2+bx+c, the discriminant, D is b2-4ac. Write a function to compute the discriminant, D.

**Examples**

>>> quadratic(1, 2, 3)

'The discriminant is -8.'

>>> quadratic(1, 3, 2)

'The discriminant is 1.'

>>> quadratic(1, 4, 4)

'The discriminant is 0.'

def quadratic(a, b, c):

return (((b)\*\*2)-(4\*a\*c))

Topic 2: Question 13

Define a function calls addFirstAndLast(x) that takes in a list of numbers and returns the sum of the first and last numbers.

**Examples**

>>> addFirstAndLast([])

0

>>> addFirstAndLast([2, 7, 3])

5

>>> addFirstAndLast([10])

10

def addFirstAndLast(x):

if len(x)==0:

return 0

if len(x)==1:

return x[0]

return (x[0]+x[-1])

Topic 2: Question 14

*lambda* can be considered to be an anonymous and/or inline function. It takes the form of "*lambda args : expression*."

**Examples**

>>> add = **lambda** x,y:x+y

>>> add(3, 4)

7

>>> even(5)

False

>>> even(8)

True

# Complete the 'lambda' expression so that it returns True if the argument is an even number, and False otherwise.

even = lambda x: x%2==0

Topic 2: Question 15

The first string statement after a function definition is the docstring. It can be accessed by the \_\_doc\_\_ keyword.

**Examples**

>>> getScore.\_\_doc\_\_

'A function that computes and returns the final score.'

# Add in the documentation string which gives the same output shown in the example.

def getScore(data):

return score

Topic 2: Question 15

The first string statement after a function definition is the docstring. It can be accessed by the \_\_doc\_\_ keyword.

**Examples**

>>> getScore.\_\_doc\_\_

'A function that computes and returns the final score.'

# Add in the documentation string which gives the same output shown in the example.

def getScore(data):

return score

Topic 2: Question 16

In Python, it is possible to pass a function as a argument to another function. Write a function useFunction(func, num) that takes in a function and a number as arguments. The useFunction should produce the output shown in the examples given below.

**Examples**

>>> **def** addOne(x):

**return** x + 1

>>> useFunction(addOne, 4)

25

>>> useFunction(addOne, 9)

100

>>> useFunction(addOne, 0)

1

def addOne(x):

return x + 1

def useFunction(func, num):

return addOne(num)\*\*2

Topic 2: Question 17

Write a function calDistance(x1, y1, x2, y2) to calculate the distance between two points represented by Point 1 (x1, y1) and Point 2 (x2, y2). The formula for calculating distance is given below:  
     **distance = √ (*x2-x1*)2 +  (*y2-y1*)2**

**Examples**

>>> calDistance(1, 0, 0, 0)

1.0

>>> calDistance(1, 1, 1, 1)

0.0

>>> calDistance(0, 0, 1, 1)

1.4142135623730951

import math

def calDistance(x1,y1,x2,y2):

d=math.sqrt(((x2-x1)\*\*2)+((y2-y1)\*\*2))

return d