

#### **P** 4.1 Program to obtain three numbers and print their sum.

```

rogram      # to input 3 numbers and print their sum
             num1 = int( input("Enter number 1 : "))
             num2 = int( input("Enter number 2 : "))
             num3 = int( input("Enter number 3 : "))
             Sum = num1 + num2 + num3
             print("Three numbers are : ", num1, num2, num3)
             print("Sum is : ", Sum)

```

The output produced by above program is as shown below :

```

Enter number 1 : 7
Enter number 2 : 3
Enter number 3 : 13
Three numbers are : 7 3 13
Sum is : 23

```

#### **P** 4.2 Program to obtain length and breadth of a rectangle and calculate its area.

```

rogram      # to input length and breadth of a rectangle and calculate its area
             length = float( input("Enter length of the rectangle : "))
             breadth = float( input("Enter breadth of the rectangle : "))

             area = length * breadth

             print ("Rectangle specifications ")
             print ("Length = ", length, end = ' ')
             print ("breadth = ", breadth)
             print ("Area = ", area)

```

The output produced by above program is as shown below :

```

Enter length of the rectangle : 8.75
Enter breadth of the rectangle : 35.0
Rectangle specifications
Length = 8.75 Breadth = 35.0
Area = 306.25

```

#### **P** 4.3 Program to get selling price and GST rate and then print Invoice along with both CGST and SGST values.

```

""" SP: selling price  cgst:central govt gst   sgst: state govt gst """
item = input("Enter item name : ")
SP = float(input("Enter selling price of item "+ item + " : "))
gstRate = float(input("Enter GST rate (%) : "))
cgst = SP * ( (gstRate/2)/ 100 )
sgst = cgst
amount = SP + cgst + sgst # consumer will buy at this price

print ("\t INVOICE")

print("Item :", item)
print("Price :", SP)
print(" CGST (@", (gstRate/2), "%) :", cgst)
print(" SGST (@", (gstRate/2), "%) :", sgst)
print("Amount payable :", amount)

```

The output produced by above program is as shown below :

```

Enter item name : Head Phones
Enter selling price of item Head Phones : 2195
Enter GST rate (%) : 18

```

#### INVOICE

```

Item :   Head Phones
Price : 2195.0
  CGST (@ 9.0 %) : 197.55
  SGST (@ 9.0 %) : 197.55
Amount payable : 2590.1000000000004

```

## LET US REVISE

- ❖ A Python program can contain various components like expressions, statements, comments, functions, blocks and indentation.
- ❖ An expression is a legal combination of symbols that represents a value.
- ❖ A statement is a programming instruction.
- ❖ Comments are non-executable, additional information added in program for readability.
- ❖ In Python, comments begin with a # character.
- ❖ Comments can be single-line comment, multi-line comments and inline comments.
- ❖ Function is a named code that can be reused with a program.
- ❖ A block/suite/code-block is a group of statements that are part of another statement.
- ❖ Blocks are represented through indentation.
- ❖ A variable in Python is defined only when some value is assigned to it.
- ❖ Python supports dynamic typing i.e., a variable can hold values of different types at different times.
- ❖ The input() is used to obtain input from user ; it always returns a string type of value.
- ❖ Output is generated through print() (by calling print function) statement.

## Solved Problems

1. What is the difference between a keyword and an identifier ?

Solution. **Keyword** is a special word that has a special meaning and purpose. Keywords are reserved and are a few. For example, *if, elif, else* etc. are keywords.

**Identifier** is a user-defined name given to a part of a program *viz.* variable, object, function etc. Identifiers are not reserved. These are defined by the user but they can have letters, digits and a symbol underscore. They must begin with either a letter or underscore. For instance, *\_chk, chess, trial* etc. are identifiers in Python.

2. What are literals in Python ? How many types of literals are allowed in Python ?

Solution. Literals mean constants *i.e.*, the data items that never change their value during a program run. Python allows *five* types of literals :

- (i) String literals
  - (ii) Numeric literals
  - (iii) Boolean literals
  - (iv) Special Literal *None*
  - (v) Literal Collections like tuples, lists etc.
3. How many ways are there in Python to represent an integer literal ?

Solution. Python allows *three* types of integer literals :

- (a) Decimal (base 10) integer literals
- (b) Octal (base 8) integer literals
- (c) Hexadecimal (base 16) integer literals

(a) **Decimal Integer Literals.** An integer literal consisting of a sequence of digits is taken to be decimal integer literal unless it begins with 0 (digit zero).

For instance, 1234, 41, +97, -17 are decimal integer literals.

(b) **Octal Integer Literals.** A sequence of digits starting with 0o (digit zero followed by letter o) is taken to be an octal integer. For instance, decimal integer 8 will be written as 0o10 as octal integer. ( $8_{10} = 10_8$ ) and decimal integer 12 will be written as 0o14 as octal integer ( $12_{10} = 14_8$ ).

(c) **Hexadecimal Integer Literals.** A sequence of digits preceded by 0x or 0X is taken to be an hexadecimal integer.

For instance, decimal 12 will be written as 0XC as hexadecimal integer.

Thus number 12 will be written either as 12 (as decimal), 0o14 (as octal) and 0XC (as hexadecimal).

4. What will be the sizes of following constants :

'\a', "\a", "Reema\'s", '\', "it's", "XY \ , ""XY\ ,  
YZ" , ""XY\ ,  
YZ""

Solution.

- '\a'      Size is 1 as there is 1 character (escape sequence) and it is a string literal enclosed in single quotes.
- "\a"      Size is 1 as there is 1 character enclosed in double quotes.

- "Reema\'s" Size is 7 because it is a string having 7 characters enclosed in double quote (escape sequence for apostrophe and is considered a single character.)
- '\'' Size is 1. It is a character constant and is containing just one character '\'
- "it's" Size is 4. Python allows single quote without escape sequence in a double quoted string and a double quote without escape sequence in a single quoted string e.g., 'no.'1
- "xy\  
yz" Size is 4. It is a multi-line string create with \ in the basic string.
- """xy ↵  
yz""" Size is 5. Triple quoted multi-line string, EOL (↵) character is also counted in

5. How many types of strings are supported in Python ?

Solution. Python allows *two* string types :

- (i) *Single-line Strings* Strings that are terminated in single line
- (ii) *Multi-line Strings* Strings storing multiple lines of text.

6. How can you create multi-line strings in Python ?

Solution. Multi-line strings can be created in *two* ways :

- (a) By adding a backslash at the end of normal single-quote or double-quote strings e.g.,

```
Text = "Welcome \  
To\  
Python"
```

- (b) By typing the text in triple quotation marks. (No backslash needed at the end of line)

```
Str1 = """Welcome  
To  
Python  
"""
```

7. What factors guide the choice of identifiers in programs ?

Solution.

- (i) An identifier must start with a letter or underscore followed by any number of digits and/or letters.
- (ii) No reserved word or standard identifier should be used.
- (iii) No special character (other than underscore) should be included in the identifier.

8. Write the following real constants in exponent form : 17.251, 151.02, 0.00031, 0.452.

Solution.

- (i)  $17.251 = 0.17251 \times 10^2 = 0.17251E02$
- (ii)  $151.02 = 0.15102 \times 10^3 = 0.15102E03$
- (iii)  $0.00031 = 0.31 \times 10^{-3} = 0.31E-3$
- (iv)  $0.452 = 0.0452 \times 10^1 = 0.0452E01$

9. What is None literal in Python ?

Solution. Python has one special literal called **None**.

The **None** literal is used to indicate something that has not yet been created in simple words, absence of value. It is also used to indicate the end of lists in Python.

10. Identify the types of following literals :

23.789	23789	True	'True'	"True"
False	""False""	0XFACE	0o213	0o789

Solution.

23.789	Floating point
23789	integer
True	Boolean
'True'	String
"True"	String
False	Boolean
""False""	String
0XFACE	Integer (Hexadecimal)
0o213	Integer (Octal)
0o789	Invalid token ( beginning with 0 means it is octal number but digits 8 and 9 are invalid digits in octal numbers)
None	None

11. What is the difference between an expression and a statement in Python ?

Solution.

Expression	Statement
Legal combination of symbols	Programming instruction as per Python syntax
Represents something	Does something
Python evaluates it	Python executes it
End result is a value	Need not result in a value
Example :	Examples :
2.3	print ("Hello")
(3 + 5) / 4	if a > 0 :

12. Which of the following are syntactically correct strings? State reasons.

- "This course is great!"
- 'She shouted "Hello!" very loudly.'
- "Goodbye'
- 'This course is great!'
- "Hello
- "I liked the movie 'Bruce Almighty' very much."

Solution. Strings (a), (b), (d) and (f) are syntactically correct. (Strings (b) and (f) are also valid as single quote strings can use double-quotes inside them and vice versa.)

String (c) ("Goodbye' ) is incorrect because opening and closing quotes don't match.

String (e) ("Hello) is invalid because it has no closing quotes.

13. What is the error in following Python program with one statement ?

```
print ("My name is", name)
```

Suggest a solution.

Solution. The above statement is trying to print the value of an undefined variable name. The solution to above problem is to define the variable name before using it, i.e.,

```
name = 'Tanya'
print ("My name is", name)
```

14. The following code is not giving desired output. We want to input value as 20 and obtain output as 40. Can you pinpoint the problem ?

```
Number = input( "Enter Number" )
DoubleTheNumber = Number * 2
Print (DoubleTheNumber)
```

Solution. The problem is that `input()` returns value as a string, so the input value 20 is returned as string '20' and not as integer 20. So the output is not 40.

Also `Print()` is not legal function of Python ; it should be `print()`.

15. What would be the correction for problem of previous question ?

Solution. By using `int()` with `input()`, we can convert the string into integer value, i.e., as :

```
Number = int(input( "Enter Number" ) )
DoubleTheNumber = Number * 2
print (DoubleTheNumber)
```

Now the program will print desired output as 40 if 20 is typed as input.

16. Why is following code giving errors ?

```
name = "Rehman"
print ("Greetings !!!")
    print ("Hello", name)
    print ("How do you do ?")
```

Solution. The problem with above code is inconsistent indentation. In Python, we cannot indent a statement unless it is inside a suite and we can indent only as much as required.

Thus, corrected code will be :

```
name = "Rehman"
print ("Greetings !!!")
print ("Hello", name)
print ("How do you do ?")
```

17. Write a program to calculate profit percentage from the sales of goods that you made.

Solution.

```
"""
cgos : cost of goods sold
revenue : amount at which goods are sold
pl : profit or loss
"""
cgos = float(input("Enter cost of goods sold:"))
```

```

revenue = float(input("Enter revenue amount:"))
pl = revenue - cgos
prPerc = pl / cgos * 100
print("Cost of goods : ", cgos)
print("Revenue generated : ", revenue)
print("Profit % : ", prPerc)

```

18. What will be the output produced by following code ?

```

value = 'Simar'
age = 17
print (name, ", you are ", 17, "now but", end = '')
print (" you will be ", age + 1, "next year")

```

Solution.

Simar, you are 17 now but you will be 18 next year .

19. What will be the output of following code ?

```

x, y = 2, 6
x, y = y, x + 2
print (x, y)

```

Solution.

6 4

**Explanation.** First line of code assigns values 2 and 6 to variables *x* and *y* respectively.

Next line (second line) of code first evaluates right hand side *i.e.*, *y*, *x + 2* which is 6, 2 + 2 *i.e.*, 6, 4 ; and then assigns this to *x*, *y* *i.e.*, *x*, *y = 6, 4* ; so *x* gets 6 and *y* gets 4.

Third line prints *x* and *y* so the output is 6 4.

20. Predict the output of following :

```

x, y = 7, 2
x, y, x = x + 1, y + 3, x + 10
print (x, y)

```

Solution.

17 5

## GLOSSARY

<b>Constant</b>	A data item that never changes its value during a program run.
<b>Identifier</b>	Name given by user for a part of the program.
<b>Keyword</b>	Reserved word having special meaning and purpose.
<b>Lexical Unit</b>	Other name of token.
<b>Literal</b>	Constant.
<b>Token</b>	The smallest individual unit in a program.
<b>String literal</b>	Sequence of characters enclosed in any type of quotes.
<b>Variable</b>	Named stored location whose value can be manipulated during program run.
<b>Expression</b>	Legal combination of symbols that represents a value.

## Assignments

### Type A : Short Answer Questions/Conceptual Questions

1. What are tokens in Python ? How many types of tokens are allowed in Python ?  
Exemplify your answer.
2. How are keywords different from identifiers ?
3. What are literals in Python ? How many types of literals are allowed in Python ?
4. Can nongraphic characters be used in Python ? How ? Give examples to support your answer.
5. How are floating constants represented in Python ? Give examples to support your answer.
6. How are string-literals represented and implemented in Python ?
7. Which of these is not a legal numeric type in Python ? (a) int (b) float (c) decimal.
8. Which argument of print( ) would you set for :  
(i) changing the default separator (space) ? (ii) printing the following line in current line ?
9. What are operators ? What is their function ? Give examples of some unary and binary operators.
10. What is an expression and a statement ?
11. What all components can a Python program contain ?
12. What do you understand by block/code block/suite in Python ?
13. What is the role of indentation in Python ?
14. What are variables ? How are they important for a program ?
15. What do you understand by undefined variable in Python ?
16. What is Dynamic Typing feature of Python ?
17. What would the following code do : `X = Y = 7` ?
18. What is the error in following code : `X, Y = 7` ?
19. Following variable definition is creating problem `X = 0281`, find reasons.
20. "Comments are useful and easy way to enhance readability and understandability of a program".  
Elaborate with examples.

### Type B : Application Based Questions

1. From the following, find out which assignment statement will produce an error. State reason(s) for each.
 

(a) <code>x = 55</code>	(b) <code>y = 037</code>	(c) <code>z = 0o98</code>	(d) <code>56thnumber = 3300</code>
(e) <code>length = 450.17</code>	(f) <code>!Taylor = 'Instant'</code>	(g) <code>this variable = 87.E02</code>	
(h) <code>float = .17E - 03</code>	(i) <code>FLOAT = 0.17E - 03</code>		
2. Find out the error(s) in following code fragments :
 

(i) <code>temperature = 90</code> <code>print temprature</code>	(ii) <code>a = 30</code> <code>b = a + b</code> <code>print (a And b)</code>
(iii) <code>a, b, c = 2, 8, 9</code> <code>print (a, b, c)</code> <code>c, b, a = a, b, c</code> <code>print (a ; b ; c)</code>	(iv) <code>X = 24</code> <code>4 = X</code>
(v) <code>print ("X = " X)</code>	(vi) <code>else = 21 - 5</code>



3. What will be the output produced by following code fragment (s) ?

```
(i) X = 10
    X = X + 10
    X = X - 5
    print (X)
    X, Y = X - 2, 22
    print (X, Y)
```

```
(ii) first = 2
     second = 3
     third = first * second
     print (first, second, third)
     first = first + second + third
     third = second * first
     print (first, second, third)
```

```
(iii) side = int(input('side' ) )    #side given as 7
      area = side * side
      print (side, area)
```

4. What is the problem with the following code fragments ?

```
(i) a = 3
    print (a)
    b = 4
    print (b)
    s = a + b
    print (s)
```

```
(ii) name = "Prejith"
     age = 26
     print ("Your name & age are ", name + age)
```

```
(iii) a = 3
      s = a + 10
      a = "New"
      q = a / 10
```

5. Predict the output :

```
x = 40
y = x + 1
x = 20, y + x
print (x, y)
```

6. Predict the output

```
x, y = 20, 60
y, x, y = x, y - 10, x + 10
print (x, y)
```

7. Predict the output

```
(a) a, b = 12, 13
    c, b = a*2, a/2
    print (a, b, c)
```

```
(b) a, b = 12, 13
    print (print(a + b))
```

8. Predict the output

```
a, b, c = 10, 20, 30
p, q, r = c - 5, a + 3, b - 4
print ('a, b, c :', a, b, c, end = '')
print ('p, q, r :', p, q, r)
```

9. Find the errors in following code fragment

```
(a) y = x + 5          (b) print (x = y = 5)      (c) a = input("value")
    print (x, Y)                                     b = a/2
                                                    print (a, b)
```

10. Find the errors in following code fragment : (The input entered is XI)

```
c = int(input("Enter your class"))
print ("Your class is", c)
```

11. Consider the following code :

```
name = input("What is your name?")
print ('Hi', name, ',')
print ("How are you doing?")
```

was intended to print output as

```
Hi <name>, How are you doing ?
```

But it is printing the output as :

```
Hi <name>,
How are you doing?
```

What could be the problem ? Can you suggest the solution for the same ?

12. Find the errors in following code fragment :

```
c = input("Enter your class")
print ("Last year you were in class") c - 1
```

13. What will be returned by Python as result of following statements?

```
(a) >>> type(0)          (b) >>> type(int(0))          (c) >>>.type(int('0'))
(d) >>> type('0')       (e) >>> type(1.0)          (f) >>> type(int(1.0))
(g) >>>type(float(0))   (h) >>> type(float(1.0)) (i) >>> type(3/2)
```

Match your result after executing above statements.

14. What will be the output produced by following code ?

```
(a) >>> str(print())+"One"
(b) >>> str(print("hello"))+"One"
```

Match your result after executing above statements. Can you explain why this result came?

15. What will be the output produced by following code ?

```
(a) >>> print(print("Hola"))
(b) >>> print(print("Hola", end = " "))
```

Match your result after executing above statements. Can you explain why this result came?

16. Carefully look at the following code and its execution on Python shell. Why is the last assignment giving error ?

```
>>> a = 0o12
>>> print(a)
10
>>> b = 0o13
>>> c = 0o78
File "<python-input-41-27fbe2fd265f>", line 1
  c = 0o78
    ^
```

SyntaxError : invalid syntax

17. Predict the output

```
a, b, c = 2, 3, 4
a, b, c = a*a, a*b, a*c
print(a, b, c)
```

18. The `id()` can be used to get the memory address of a variable. Consider the following code and tell if the `id()` functions will return the same value or not (as the value to be printed via `print()`) ? Why ?  
[There are four `print()` function statements that are printing id of variable *num* below]

```
num = 13
print( id(num) )
num = num + 3
print( id(num) )
num = num - 3
print( id(num) )
num = "Hello"
print( id(num) )
```

19. Consider below given two sets of codes, which are nearly identical, along with their execution in Python shell. Notice that first code-fragment after taking input gives error, while second code-fragment does not produce error. Can you tell why ?

(a)

```
>>> print(num = float(input("value1:"))) )
value1:67 ← Input taken as per execution of input()
Traceback (most recent call last):
  File "<python-input-56-78b83d911bc6>", line 1, in <module>
    print(num = float(input("value1:"))) )
TypeError: 'num' is an invalid keyword argument for this function
```

(b)

```
>>> print(float(input("value1:"))) )
value1:67
67.0 ← Code successfully executed. No error reported.
```

20. Predict the output of the following code :

```
days = int(input("Input days: ")) * 3600 * 24
hours = int(input("Input hours: ")) * 3600
minutes = int(input("Input minutes: ")) * 60
seconds = int(input("Input seconds: "))
time = days + hours + minutes + seconds
print("The amounts of seconds", time)
```

If the input given is in this order : 1, 2, 3, 4

### Type C : Programming Practice/Knowledge based Questions

- Write a program that displays a joke. But display the punchline only when the user presses enter key.  
(Hint. You may use input( ))
- Write a program to read today's date (only del part) from user. Then display how many days are left in the current month.
- Write a program that generates the following output :
 

```
5
10
9
```

Assign value 5 to a variable using assignment operator (=) Multiply it with 2 to generate 10 and subtract 1 to generate 9.
- Modify above program so as to print output as 5@10@9.
- Write the program with maximum three lines of code and that assigns first 5 multiples of a number to 5 variables and then print them.
- Write Python program that accepts marks in 5 subjects and outputs average marks.
- Write a short program that asks for your height in centimetres and then converts your height to feet and inches. (1 foot = 12 inches, 1 inch = 2.54 cm).
- Write a program to read a number  $n$  and print  $n^2$ ,  $n^3$  and  $n^4$ .
- Write a program to compute simple interest and compound interest.
- Write a program to input a number and print its first five multiples.
- Write a program to read details like *name*, *class*, *age* of a student and then print the details firstly in same line and then in separate lines.  
Make sure to have two blank lines in these two different types of prints.
- Write a program to read three numbers in three variables and swap first two variables with the sums of first and second, second and third numbers respectively.
- Write a program that accepts cost of goods sold (cgos) revenue generated, operating costs (oc) and prints Gross profit, net profit and net profit percentage.  
[Hint : Net profit = Revenue - cgos - oc]