

## PYTHON

1. State and explain the primitive data type of python

Ans - The data type of a variable or object determines which operations can be applied to it. Once a variable is assigned a data type, it can be used for computations in the program.  
 primitive data type :- Data types which are pre-defined and supported by the programming language.

(i) Integer -

Integers represent numeric data - they represent whole numbers from negative infinity to infinity

eoc - `integer_1 = 100`

`integer_2 = 50`

# various operations on integers

`print (integer_1 * integer_2)`

`print (integer_1 + integer_2)`

`print (integer_1 - integer_2)`

`print (integer_1 / integer_2)`

(ii) string -

strings are used to store data that involves characters (e.g. names of address).

strings can be created in python by enclosing a sequence of characters within a pair of single or double-quoted.

eoc - `string_1 = "Hello"`

`string_2 = "world"`

`print (string_1 + string_2)`

(iii) **Float -**

Floating points are used for rational numbers that usually end with a decimal number.

Ex -  $\text{float\_1} = 12.539$

$\text{float\_2} = 6.78$

# Various operations on float

`print (float_1 * float_2)`

`print (float_1 + float_2)`

`print (float_1 - float_2)`

`print (float_1 / float_2)`

2. State and compare implicit and explicit type conversion with examples for each

Ans - Implicit - Implicit conversion do not require any special syntax.

- \* Implicit conversion is done automatically
- \* In Implicit conversion, no data loss take place during the data conversion
- \* No possibility of throwing exception during the conversion and therefore it called type safe.

Ex :-

conversion of integer type data to float

`float i = 0;`

`int j = 10;`

`i = j;`

Explicit - Explicit conversion do require cast operator to perform conversion

- \* In explicit conversion, data loss may or may not be take place during data conversion. Hence there is a risk of information loss.
- Ex - float k = 123.456  
 $\text{print}(k) = \text{print}(k)$

3. Illustrate how input and output is achieved in python with simple example program

Ans -

```
# Taking input from the user
name = input("Enter your name:")
# output
print("Hello, " + name)
print(type(name))
```

Output:

```
Enter your name: GFG
Hello, GFG
<class 'str'>
```

python takes all the input as a string input by default. To convert it to any other data type we have to convert the input explicitly.

4. With syntax illustrate the if-else construct with example

Ans - Syntax -

```
if conditional-expression:
    clause
else:
    clause
```

# program to check whether a given no is +ve or -ve

```
x = int(input("Enter an integer number"))
if (x < 0):
    print('The number', x, 'is negative')
else:
    print('The number', x, 'is positive')
```

The execution has to execute only one condition expression at line-3 if its evaluation is true, line-4 will be executed, if the condition is false line 6 will be executed.

5. write a program to check whether a given number is positive/negative/zero with appropriate messages.

```
AN
x = int(input("Enter any number"))
if (x == 0):
    print("The number", x, "is zero")
elif (x < 0):
    print("The number", x, "is negative")
else:
    print("The number", x, "is positive")
```

%/P

Enter any number: 3

The number 3 is positive

6. write the output of the following code snippet

$x, y = 5, 6$

if ( $x > y$ ) is true:

    print ("This is first output line")

else:

    print ("This is second output line")  
output -

This is first output line

8. Write a program to compute the grade of a student given his marks in 4 subjects. Assume minimum marks & 35 & max & 100. Display appropriate message.

Ans -

a = float (input ("Enter the marks of 1<sup>st</sup> subject"))

b = float (input ("Enter the marks of 2<sup>nd</sup> subject"))

c = float (input ("Enter the marks of 3<sup>rd</sup> subject"))

d = float (input ("Enter the marks of 4<sup>th</sup> subject"))

avg = (a+b+c+d)/4

if (a < 35 or b < 35 or c < 35 or d < 35):

    print ("The result is fail")

elif (avg >= 100):

    print ("The result is distinction", avg)

elif (avg >= 75):

    print ("The result is first class", avg)

elif (avg >= 55):

    print ("The result is second class", avg)

else:

    print ("The result is third class", avg)

endif

Enter the marks of 1<sup>st</sup> subject : 80

Enter the marks of 2<sup>nd</sup> subject : 67

Enter the marks of 3<sup>rd</sup> subject : 90

Enter the marks of 4th subject: 92  
 The Result is first class 82.025

Q. With syntax explain how while loop works with a sample program

Ans - Syntax:

Initialization statement

while (condition-expression):  
 block of statements

Example: program to add natural numbers up to n

# n = Ent (Input ("Enter the limit:"))

n = 10

sum = 0

i = 1

while i <= n:

sum = sum + i

i = i + 1

print ("The sum is", sum)

O/P

Enter n: 10

The sum is 55

In the program the test condition will be true as long as our counter variable i is less than or equal to n (10 in our program)

We need to increase the value of the counter variable in the body of the loop. This is very important, failing to do so will result in an infinite loop.

10. Write a program to check whether a given number is palindrome or not

Ans -  $x = \text{int}(\text{input}("Enter the number:"))$

$n = x$

$rev = 0$

while ( $x > 0$ ):

$p = x \% 10$

$x = x // 10$

$rev = rev * 10 + p$

If ( $rev == n$ ):

print ("The number",  $n$ , "is palindrome")

else:

print ("The number",  $n$ , "is not palindrome")

Output:

Enter the number : 121

The number 121 is palindrome

11. Fill out the missing statement of the program to print Fibonacci series of  $n$  terms.

Ans -  $f, s = 0, 1$

$n = \text{int}(\text{input}("Enter the number:"))$

$q = 3$

print ( $f$ ), print ( $s$ )

while ( $i < n + 1$ ):

$t = f + s$

print ( $t$ )

$f = s$

$s = t$

$i = i + 1$

Alter the above program to print the Fibonacci series till  $n$

```
n = int(input("Enter the value :"))
```

```
f = 0
```

```
s = 1
```

```
sum = 0
```

```
Count = 1
```

```
print("Fibonacci series : ", end = " ")
```

```
while (Count <= n):
```

- print(sum, end = " ")

```
Count += 1
```

```
f = s
```

```
s = sum
```

```
sum = f + s
```

Q. State function and types of functions

Ans Built-in functions like len(), str() etc.  
User-defined functions can also be created. Functions always help to avoid duplicating code. A major purpose of functions is to group code that gets executed multiple times.

3 types of functions

- \* Built-in functions, such as help() to ask for help, min() to get the minimum value, print() to print an object to the terminal, etc.
- \* User-defined functions which are functions that we create to help them out.
- \* Anonymous functions, which are also called lambda functions because they are not

declared with the standard def' keyword.

13. Write a program to check whether a given number is prime or not using function that returns 1 if the given number is prime else return 0

Ans - def prime(x):

p = 0

if x > 1:

for i in range(2, x):

if (x % i == 0):

return 1

return 0

x = int(input("Enter the number:"))

p = prime(x)

if (p):

return 0

else:

return 1

14. Differentiate b/w Local & global variable with example for each

Ans - Local variable

- \* It is declared inside a function
- \* If it is not initialized, a garbage value is stored
- \* It is created when the function starts execution & lost when the function terminates

Global variable

- \* It is declared outside the function
- \* If it is not initialized, zero is stored as default
- \* It is created before the program's global execution starts & lost when the program terminates

- |  |   |
|--|---|
| * parameters passing<br>is required for local<br>variables to access the<br>values in other function | * parameters passing<br>is necessary for a<br>global variable as it<br>is visible throughout<br>the program |
| * It is stored on the<br>stack which is specified  | * It is stored on a fixed<br>location decided by<br>the compiler  |

local variables - example

```
def sum(x,y):
    sum = x + y
    return sum
print(sum(5,10))
O/P
5
```

global variable - example

$z = 25$

```
def function():
    global z
    print(z)
    z = 20
function()
print(z)
```

O/P  
25

20

15. Write a program to generate 10 random numbers between 15 & 35

Ans `import random`

`x, y = 15, 35`

`for i in range(1, 11):`

`print(random.randint(x, y))`

16. State exception handling in python with syntax and example

Ans- Try and except statements are used to catch and handle exceptions in python. Statements that can raise exceptions are kept inside the try clause and the statements that handle the exception are written inside except clause

(d)

Exceptions are raised when the program is syntactically correct, but the code resulted in an error. The error does not stop the execution of the program, however, it changes the normal flow of the program.

Syntax :

try :

block of code that may raise exception

except name - q - exception :

action to perform in case of exception

Ex-

`a=5`

`b=10`

`c=b-a*a`

`try:`

`print(a/c)`

except :

print ("Division by zero")

17. Write a program to count the even digits present in given number, using for loop

Ans - n = int(input("Enter the number"))

x = str(n)

Count = 0

for i in x:

i = int(i)

if (i % 2 == 0):

count = count + 1

print("There are", count, "even numbers in a given number")

18. what is list, mention the syntax with an example

Ans A list is a data structure in python that is a mutable, or changeable, ordered sequence of elements. To create a list, the elements are placed inside square brackets ([]), separated by commas.

Syntax :

list<data\_type> = Array

Example :

a = ['p', 'g', 'o', 'b', 'e']

print(a[-1])

# last item

print(a[-5])

# fifth last item

O/P  
e

P

19. Write a program to count number of occurrences of a character in given string.

Ans n = input("Enter the string")

m = input("Enter the alphabet to count")

Count = 0

for i in n:

if (i == m):

Count = Count + 1

print(m, "occurred", Count, "times")

Output:

20. Write a program to accept the list by input

L1 = [100, "Rahul", 101, "Praya", "Karan",  
102, 103, "Amar"]

Ans L1 = [100, "Rahul", 101, "Praya", "Karan", 102, 103,  
"Amar"]

L2 = []

L3 = []

for i in L1:

if (type(i) == int):

L2.append(i)

else:

L3.append(i)

print(L2)

print(L3)

7. Find out the syntax error in the following code & correct them

a, b, c d = 1, 2, 3, 4

if(a+b)>5 && (c>d)

print("The will be printed from a block")

else:

print("This will be printed from else block")

Ans The syntax error in the program is that the representation of a logical operator.

In python symbols like & || are not used for representing logical operators instead of that keyword 'and' 'or' is used  
a,b,c,d=1,2,3,4

If (a+b)>5 and (c<d):

print("This will be printed from if block")

else:

print("This will be printed from else block")

There is also error in "if" syntax i.e., the if statement syntax is  
if (condition):

but in the given program colon (:) is not mentioned