

Python Solved Question Paper

Q.1 State and explain any six applications of python

- Web Development
- Game Development
- Data Science and Data Visualization
- Desktop GUI
- Web Scraping Applications
- Business Applications
- Audio and Video Applications
- CAD Applications
- Embedded Applications

Explain

- Web Development = Python can be used to make web-applications at a rapid rate. Some of the most well-known frameworks are Django, Flask, Pyramid. Why use a framework? The **security, scalability, convenience**
- Game Development = Python is also used in the development of interactive games. Games such as Civilization-IV, Disney's Toontown Online, Vega Strike etc. have been built using Python.
- Data Science and Data Visualization = Data is *money* if you know how to extract relevant information which can help you take calculated risks and increase profits. You study the data you have, perform operations and extract the information required. Libraries such as Pandas, NumPy help you in extracting information.
- Desktop GUI = We use Python to program **desktop applications**. You can start out with creating simple applications such as Calculators, To-Do apps and go ahead and create much more complicated applications
- Business Applications = Business Applications are different than our normal applications covering domains such as e-commerce, ERP and many more. They require applications which are scalable, extensible and easily readable
- Audio and Video Applications = We use Python to develop applications that can multi-task and also output media. They provide better stability and performance in comparison to other media players.

State which of the following python statements are valid and invalid:

a. `x = input("enter your name")`

b. `x = input()`

c. `x = input("")`

d. `x = input("""`

e. `x = input(2)`

f. `x = input("python" + 53)`

ENGG SOLUTION

What are identity operators; explain With suitable examples

Operator	Description	Example
is	Evaluates to true if the variables on either side of the operator point to the same object and false otherwise.	x is y, here is results in 1 if id(x) equals id(y).
is not	Evaluates to false if the variables on either side of the operator point to the same object and true otherwise.	x is not y, here is not results in 1 if id(x) is not equal to id(y).

```
a = 20
```

```
b = 20
```

```
if ( a is b ):
```

```
    print "Line 1 - a and b have same identity"
```

```
else:
```

```
    print "Line 1 - a and b do not have same identity"
```

Output -

```
Line 1 - a and b have same identity
```

Q.2 Differentiate between List, Tuple & set with suitable examples of each.

Parameters	List	Tuple	Set
Basics	A list is basically like a dynamically sized array that gets declared in other languages (Arraylist in the case of Java, vector in the case of C++).	The tuples refer to the collections of various objects of Python separated by commas between them.	The sets are an unordered collection of data types. These are mutable, iterable, and do not consist of any duplicate elements.

Example	[6, 7, 8, 9, 10]	(6, 7, 8, 9, 10)	{6, 7, 8, 9, 10}
---------	------------------	------------------	------------------

Representation	We can represent a List by []	We can represent a Tuple by ()	We can represent a Set by {}
Duplicate elements	It allows various duplicate elements.	It allows various duplicate elements.	It does not allow any duplicate elements.
Nested Among All	It can be utilized in a List.	It can be utilized in a Tuple.	It can be utilized in a Set.

Explain any six assignment operators of python with suitable examples of each.

Operator	Description	Syntax
----------	-------------	--------

=	Assign value of right side of expression to left side operand	x = y + z
---	---	-----------





```
# Assigning values using  
# Assignment Operator  
a = 3  
b = 5  
c = a + b  
  
# Output  
print(c)
```

ENGG SOLUTION

Output:

8





+=	Add and Assign: Add right side operand with left side operand and then assign to left operand	a += b
----	---	--------

```
 a = 3  
b = 5  
  
 # a = a + b  
a += b  
 # Output  
print(a)
```

Output:

8

-- Subtract AND: Subtract right operand from left operand and then assign to left operand: True if both operands are equal a -= b





```
 a = 3  
b = 5  
  
 # a = a - b  
a -= b  
 # Output  
print(a)
```

ENGG SOLUTION

Output:

-2





*= Multiply AND: Multiply right operand with left operand and then assign to left operand a *= b

```
 a = 3  
 b = 5  
 # a = a * b  
a *= b  
 # Output  
print(a)
```

Output:

15

>>= Performs Bitwise right shift on operands and assign value to left operand a >>= b





```
 a = 3  
 b = 5  
 # a = a >> b  
a >>= b  
 # Output  
print(a)
```

ENGG SOLUTION

Output:

0

<<= Performs Bitwise left shift on operands and assign value to left operand a <<= b

```
 a = 3  
 b = 5  
 # a = a << b  
a <<= b  
 # Output  
print(a)
```

Output:

```
96
```

If a four-digit number is input through the keyboard, write a program (using for loop) to display sum of all digits.

ENGG SOLUTION

Calculate the sum of the digits
in an integer

5245

5 + 2 + 4 + 5

Sum of the digits

16

Python Code:

```
1 num = int(input("Input a four digit numbers: "))
2 x = num //1000
3 x1 = (num - x*1000)//100
4 x2 = (num - x*1000 - x1*100)//10
5 x3 = num - x*1000 - x1*100 - x2*10
6 print("The sum of digits in the number is", x+x1+x2+x3)
```

Sample Output:

```
Input a four digit numbers: 5245
The sum of digits in the number is 16
```

Q.3 A) If the ages of Raj, Ketan and kunal are input through the keyboard, Write program to determine the

a) Youngest of the three,

b) Oldest of three.

```
age1=int(input('enter age1:15 '))
```

```
age2=int(input('enter age2:25 '))
```

```
age3=int(input('enter age3:20'))
```

```
if age1>age2 and age1>age3:
```

```
    print('age1 is oldest')
```

```
elif age2>age1 and age2>age3:
```

```
    print('age2 is oldest')
```

```
else:
```

ENGG SOLUTION

```
    print('age3 is oldest')
```

```
if age1<age2 and age1<age3:
```

```
    print('age1 is youngest')
```

```
elif age2<age1 and age2<age3:
```

```
    print('age2 is youngest')
```

```
else:
```

```
    print('age3 is youngest')
```

Output:

```
enter age1:15 enter age2:25 enter age3:20
```

```
age2 is oldest
```

```
age1 is youngest.
```

B) Explain count() methods of list. write python program which will take name and subject as input from students (from class of 75 students) and at the end it should display names of all students and total number of students who opted for subject Python Programming.

The count() method **returns the number of times the specified element appears in the list.**

ENGG SOLUTION

C) - Explain for loop with suitable example?

The for loop in Python is used to iterate over a sequence, which could be a [list](#), [tuple](#), [array](#), or [string](#).

Syntax: FOR COUNTER IN SEQUENCE:

STATEMENT(S)

```
In [7]: #iterate through a list
        colors = ["red", "green", "yellow", "black"]
        for x in colors:
            print(x)

        #iterate through a string
        s = "red"
        for x in s:
            print(x)
```

red
green
yellow
black
r
e
d

Q.4 Explain concept of OOP in python also concept of inheritance in details

In Python, object-oriented Programming (OOPs) is a programming paradigm that uses objects and classes in programming. It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming. The main concept of OOPs is to bind the data and the functions that work on that together as a single unit so that no other part of the code can access this data.

Main Concepts of Object-Oriented Programming (OOPs)

- Class
- Objects
- Polymorphism
- Encapsulation
- Inheritance
- Data Abstraction



- **Inheritance** = Inheritance is the capability of one class to derive or inherit the properties from another class. The class that derives properties is called the derived class or child class and the class from which the properties are being derived is called the base class or parent class

B) Write python program to print following pattern
(using loop statement)

```
n=int(input())
for i in range(n):
    for j in range(i):
        print(end=" ")
    for j in range(0,n-i):
        print("*",end=" ")
    print()
```

ENGG SOLUTION

C) If any integer is input through the keyboard, Write a python program to find out whether it is a prime number or non-prime number.

```
# taking input from user
```

```
number = int(input("Enter any number: "))

# prime number is always greater than 1
if number > 1:
    for i in range(2, number):
        if (number % i) == 0:
            print(number, "is not a prime number")
            break
    else:
        print(number, "is a prime number")

# if the entered number is less than or equal to 1
# then it is not prime number
else:
    print(number, "is not a prime number")
```

Output:

```
Enter any number: 13
13 is a prime number

Process finished with exit code 0
```

Q.5 A) What is turtle in Python?

turtle is a **pre-installed Python library that enables users to create pictures and shapes by providing them with a virtual canvas**. The onscreen pen that you use for drawing is called the turtle and this is what gives the library its name

ENGG SOLUTION

B) A program has to be written in python which will take two numbers as input from user and perform the division in such a way that program should not terminate (because of ZeroDivisionError) even if user input is zero.

```
def divide_two_numbers():  
  
    try:  
        a = input("Enter an integer: ")  
        b = input("Enter another integer: ")  
  
        c = int(a)/int(b)  
  
    except ZeroDivisionError:  
        c = "You cannot divide by 0."  
  
    print(c)  
  
divide_two_numbers()
```

```
Enter an integer: 9  
Enter another integer: 0  
You cannot divide by 0.
```

In this example, we have told the user that they cannot divide by 0 if they attempt to do so.

C) If any two numbers are the input by user, write a program on arithmetic calculator (addition, subtraction, multiplication and division) with the help of OOP concepts.

```
# Python program to perform Addition Subtraction Multiplication
# and Division of two numbers

num1 = int(input("Enter First Number: "))
num2 = int(input("Enter Second Number: "))

print("Enter which operation would you like to perform?")
ch = input("Enter any of these char for specific operation +,-,*,/:")

result = 0
if ch == '+':
    result = num1 + num2
elif ch == '-':
    result = num1 - num2
elif ch == '*':
    result = num1 * num2
elif ch == '/':
    result = num1 / num2
else:
    print("Input character is not recognized!")

print(num1, ch , num2, ":", result)
```

Output 1: Addition

```
Enter First Number: 100
Enter Second Number: 5
Enter which operation would you like to perform?
Enter any of these char for specific operation +,-,*,/: +
100 + 5 : 105
```

Output 2: Division

```
Enter First Number: 20
Enter Second Number: 5
Enter which operation would you like to perform?
Enter any of these char for specific operation +,-,*,/: /
20 / 5 : 4.0
```

Output 3: Subtraction

```
Enter First Number: 8
Enter Second Number: 7
Enter which operation would you like to perform?
Enter any of these char for specific operation +,-,*,/: -
8 - 7 : 1
```

Output 4: Multiplication

```
Enter First Number: 6
Enter Second Number: 8
Enter which operation would you like to perform?
Enter any of these char for specific operation +,-,*,/: *
6 * 8 : 48
```

DONE.....



ENGG SOLUTION