1) Print bellow statement.

```
Print(“hello welcome to bca”)
```

2) Print following variables.

```
x = 5
y = "John"
print(x)
print(y)
```

3) Create a sequence of number using range data type to display 1 to 30 with an increment of 2.

```
start = 1
stop = 30
step = 2

print(list(range(start, stop, step)))
```

4) Write a program to find out and display the common and the non common elements in the list using membership operator.

```
list1=[1,3,4,5,6,8]

list2=[1,2,3,7,6,9]

for a in list1:

   if ( a in list2 ):

      print("Common number in list are:",a)

   else:

      print("Non Common Number in list are:",a)
```

5) Create a program to display memory location of two variable using id() function, and then use identity operator two compare whether two objects are same or not.

```
a = 10
b = 10

print("Memory location of a:",id(a))
print("Memory location of b:",id(b))

if ( a is b ):
  print("a and b have same identity")
else:
  print("a and b do not have same identity")

if ( id(a) == id(b) ):
  print("a and b have same identity")
else:
  print("a and b do not have same identity")
```

6) Write a program that evaluates an expression given by the user at run time using eval() function.

Example: Enter and expression: 10+8-9*2-(10*2)

Result -20

```
a = input("Enter Math Function to Evaluate:")

b = eval(a)
```

```
print ("The result of you Enter Maths function is :",b)
```

7) write a python program to find the sum of even number using command line argument

```
import sys
n=int(sys.argv[1])
i=2
sum=0
while(i<=n):
    sum=sum+i
    i=i+2
print("sum of even number=", sum)
```

9) write a menu driven python program which perform the following

Find area of circle

Find area of triangle

Find simple interest

```
def mainmenu():
   print("1: Find area of Circle")
   print("2: Find are of tringle")
   print("3: Find Simple Interest")
   print("4. Quit")
   selection=int(input("Enter your choice"))
   if selection==1:
        rad = input("Enter radius of circle: ");
        radius = float(rad);
        area = 3.14 * radius * radius;
        print("\nArea of Circle = %0.2f" %area);
        print("\n")

        mainmenu()

    elif selection==2:


        side1 = input("Enter length of first side: ");
        side2 = input("Enter length of second side: ");
        side3 = input("Enter length of third side: ");

        a = float(side1);
        b = float(side2);
        c = float(side3);
        s = (a + b + c)/2;
        area = (s*(s-a)*(s-b)*(s-c)) ** 0.5;
        print("\nArea of Triangle = %0.2f" %area);

        print("\n")
        mainmenu()

    elif selection==3:

        principle=float(input("Enter the principle amount:"))
        time=int(input("Enter the time(years):"))
        rate=float(input("Enter the rate:"))
        simple_interest=(principle*time*rate)/100
        print("The simple interest is:",simple_interest)

        print("\n")
        mainmenu()

    elif selection==4:
        exit
    else:
        print("invalid choice")
        mainmenu()

mainmenu()
```

10) Print below statement.

```
x = "awesome"
print("Python is " + x)
```

11) Add following variable using third variable.

```
x = "Python is "
y = "awesome"
z =  x + y
print(z)
```

12) Python Program to Add Two Numbers:

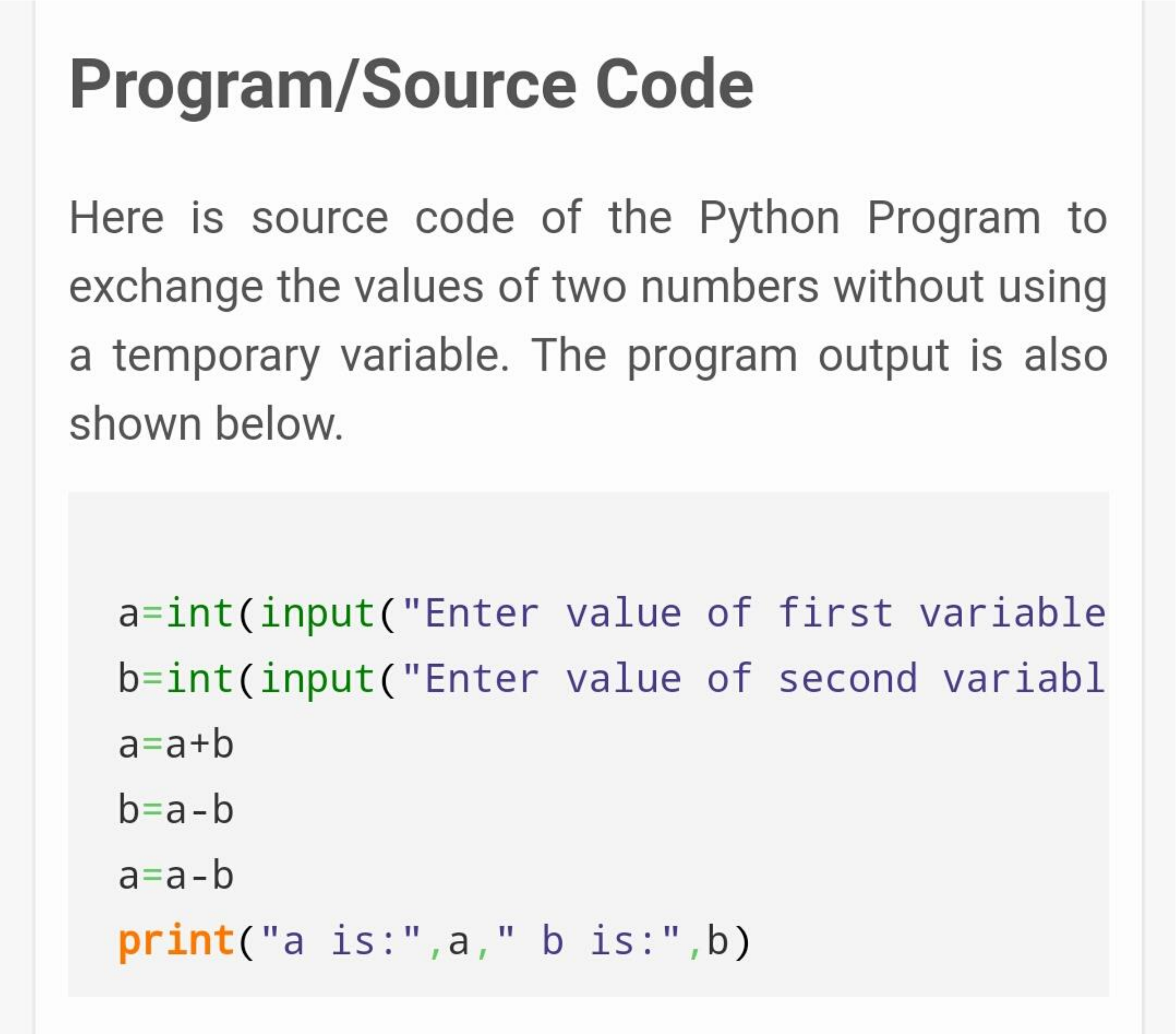
```
num1 = 1.5

num2 = 6.3
```

```
sum = float(num1) + float(num2)

print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

13) Add Two Numbers Provided by The User:

```
num1 = input('Enter first number: ')

num2 = input('Enter second number: ')

sum = int(num1) + int(num2)

print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

14) Python Program to exchange the values of two numbers without using temporary variable.



Program/Source Code

Here is source code of the Python Program to exchange the values of two numbers without using a temporary variable. The program output is also shown below.

```
a=int(input("Enter value of first variable
b=int(input("Enter value of second variabl
a=a+b
b=a-b
a=a-b
print("a is:",a," b is:",b)
```

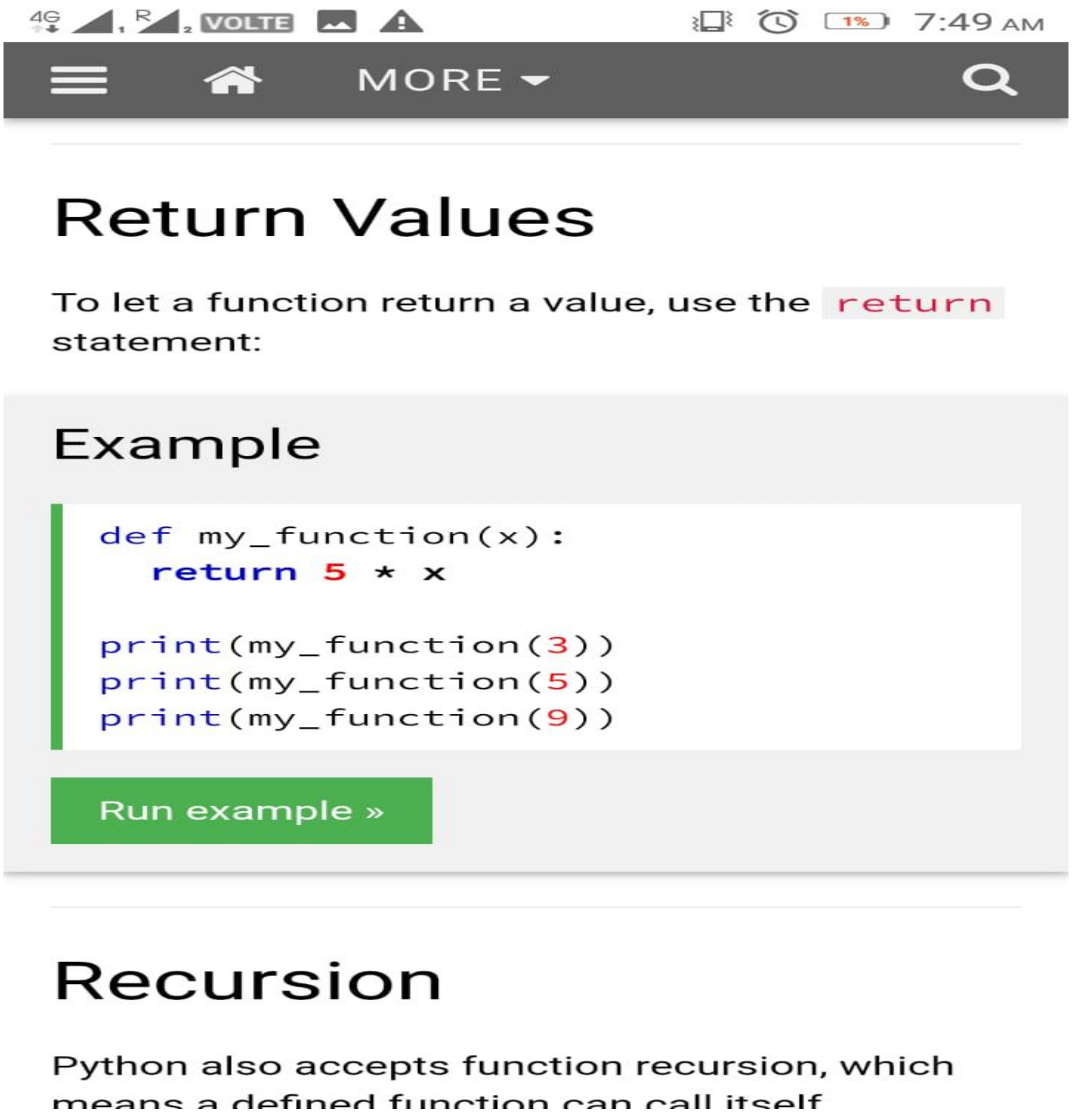
15) Python program for while loop:

```
count=0

while (count<3):

    count=count+1

    print("hello bca")
```

16) Python program for function.

```
def my_function():
    print("hello students")


my_function()
```

17) Python program for function with return value:



Return Values

To let a function return a value, use the `return` statement:

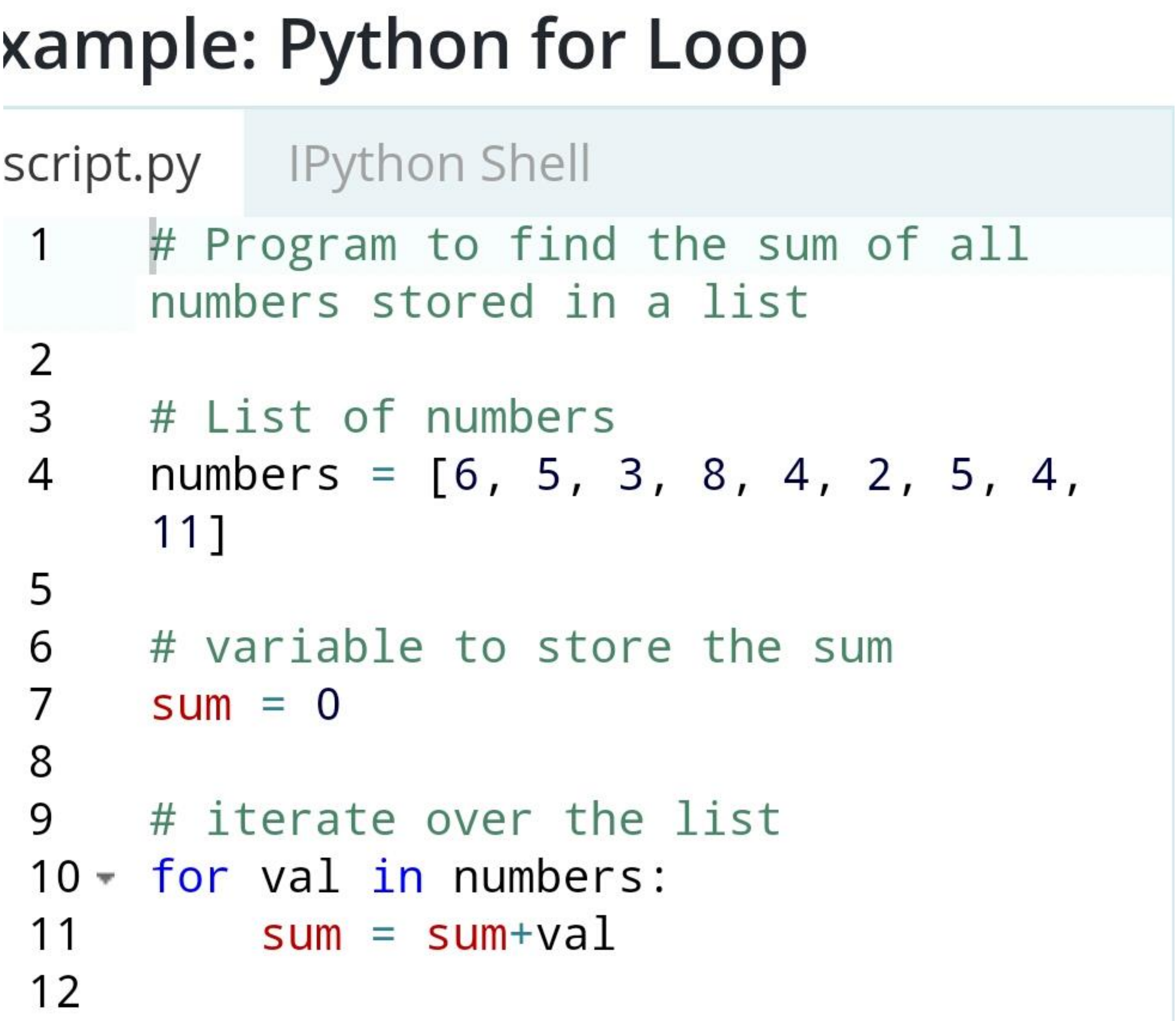
Example

```
def my_function(x):
  return 5 * x

print(my_function(3))
print(my_function(5))
print(my_function(9))
```

Recursion

Python also accepts function recursion, which means a defined function can call itself

18) Python program to find sum of all numbers stored in a list.



xample: Python for Loop

```python
# Program to find the sum of all numbers stored in a list

# List of numbers
numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]

# variable to store the sum
sum = 0

# iterate over the list
for val in numbers:
    sum = sum+val

```

19) Write a program to display sum of two complex numbers.

print("Addition of two complex numbers : ",(4+3j)+(3-7j))

20) Example of FOR loop with break statement.

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
  if x == "banana":
    break
```

21) Create a sequence of numbers using range data type to display 1 to 30, with an increment of 2.

```
print("range function example")

print("Printing range function result")

for i in range(1,30,2):

   print(i, end=', ')
```

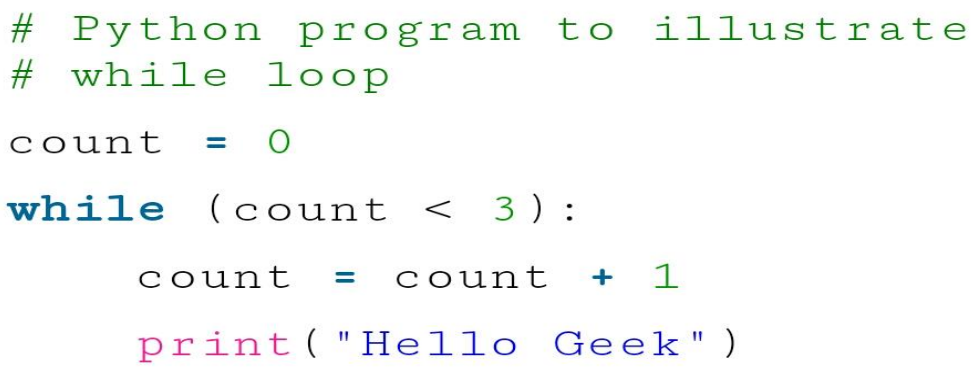
22) Python program for elif keyword:

```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

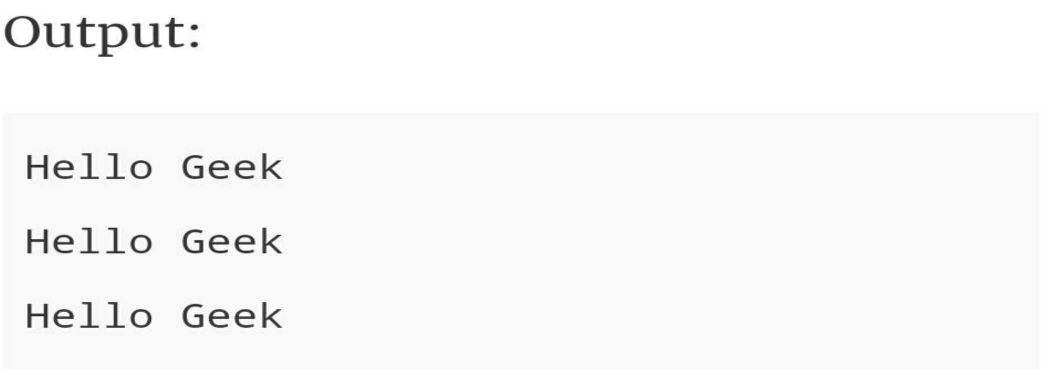
23) Python program for else keyword:

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

24) Python program to illustrate while loop.



```
# Python program to illustrate
# while loop
count = 0
while (count < 3):
    count = count + 1
    print("Hello Geek")
```



Output:

```
Hello Geek
Hello Geek
Hello Geek
```

25) Write a python program to find the sum of even numbers using command line arguments.

```
num = int(input("Enter a number: "))

mod = num % 2

if mod == 0:

   print("This is an even number.")

else:
    print("This is an odd number.")
```

26) Program to find the sum of all numbers stored in a list

```
numbers = [10, 5, 13, 28, 34, 12, 55, 14, 11]

sum = 0

for val in numbers:

        sum = sum+val

print("The sum is", sum)
```

The range() function

We can generate a sequence of numbers using range() function. range(10) will generate numbers from 0 to 9 (10 numbers).

We can also define the start, stop and step size as range(start,stop,step size). step size defaults to 1 if not provided.

- print(list(range(10)))

Output: **[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]**

- print(list(range(2,8)))

Output: [2, 3, 4, 5, 6, 7]

- **print(list(range(2, 20, 3)))**
- **Output: [2, 5, 8, 11, 14, 17]**

27) Program to print first five natural numbers:

```
sum = 0
for val in range(1, 6):
   sum = sum + val
print(sum)
```

28) Program to print squares of all numbers present in a list

```
numbers = [1, 2, 4, 6, 11, 20]

sq = 0

for val in numbers:
   sq = val * val
   print(sq)
```

29) Simple Python Program To Calculate Salary of Employee with their certain Taxable amount.

```
e_name=input("Enter the name of Employee \n")

c_name=input("Enter the company name \n")

salary=float(input("Enter the salary of Employee \n"))

if(salary>50000):
 tax=0.15*salary

 netsalary=salary-tax

 print("The net salary of "+e_name+" worked in " +c_name+ " is",netsalary)

else:

 netsalary=salary

 print("No taxalbe Amount")

 print("The net salary of "+e_name+" worked in " +c_name+ " is",salary)
```

30) Python Program to create simple calculator that can add, subtract, multiplies and divides using functions:

```
def add(x, y):
  return x + y

def subtract(x, y):
  return x - y

def multiply(x, y):
  return x * y

def divide(x, y):
  return x / y

print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

choice = input("Enter choice(1/2/3/4):")

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
```

```
if choice == '1':
  print(num1,"+",num2,"=", add(num1,num2))

elif choice == '2':
  print(num1,"-",num2,"=", subtract(num1,num2))

elif choice == '3':
  print(num1,"*",num2,"=", multiply(num1,num2))

elif choice == '4':
  print(num1,"/",num2,"=", divide(num1,num2))
else:
  print("Invalid input")
```

31)Python program to find the largest number among the three input numbers:

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

if (num1 >= num2) and (num1 >= num3):
  largest = num1
elif (num2 >= num1) and (num2 >= num3):
  largest = num2
else:
  largest = num3
```

```
print("The largest number between",num1,",",num2,"and",num3,"is",largest)
```

32)Python program to find the multiplication table (from 1 to 10)

```
# To take input from the user
num = int(input("Display multiplication table of "))

# use for loop to iterate 10 times
for i in range(1, 11):
   print(num,'x',i,'=',num*i)
```

33) Python program to display all the prime numbers within an interval.

```
lower = int(input("Enter lower range: "))
upper = int(input("Enter upper range: "))

print("Prime numbers between",lower,"and",upper,"are:")

for num in range(lower,upper + 1):
   # prime numbers are greater than 1
   if num > 1:
       for i in range(2,num):
        if (num % i) == 0:
            break
    else:
        print(num)
```

34) Python Program to find the area of triangle

```
a = float(input('Enter first side: '))

b = float(input('Enter second side: '))

c = float(input('Enter third side: '))


# calculate the semi-perimeter

s = (a + b + c) / 2


# calculate the area

area = (s*(s-a)*(s-b)*(s-c)) ** 0.5

print('The area of the triangle is %0.2f' %area)
```

35) Python Program - Calculate Area of Circle.

```
print("Enter 'x' for exit.");
rad = input("Enter radius of circle: ");
if rad == 'x':
    exit();
else:
    radius = float(rad);
    area = 3.14 * radius * radius;
    print("\nArea of Circle = %0.2f" %area);
```

---

35) Python Program - Check Alphabet or Not.

```
print("Enter '0' for exit.");
ch = input("Enter any character: ");
if ch == '0':
    exit();
else:
    if((ch>='a' and ch<='z') or (ch>='A' and ch<='Z')):
            print(ch, "is an alphabet.");
    else:
            print(ch, "is not an alphabet.");
```

---

36) Python Program - Calculate Area of Square.

```
print("Enter 'x' for exit.");
side = input("Enter side length of Square: ");
if side == 'x':
    exit();
else:
    side_length = int(side);
    area_square = side_length*side_length;
    print("\nArea of Square =",area_square);
```

---

37) Python Program - Calculate Area of Rectangle.

```
print("Enter 'x' for exit.");
```

```
leng = input("Enter length of Rectangle: ");
if leng == 'x':
   exit();
else:
   brea = input("Enter breadth of Rectangle: ");
   length = int(leng);
   breadth = int(brea);
   area = length*breadth;
   print("\nArea of Rectangle =",area);
```

38) Simple Python Program To Calculate Salary of Employee with their certain Taxable amount.

```
e_name=input("Enter the name of Employee \n")

 c_name=input("Enter the company name \n")

 salary=float(input("Enter the salary of Employee \n"))

 if(salary>50000):

 tax=0.15*salary

 netsalary=salary-tax

 print("The net salary of "+e_name+" worked in " +c_name+ " is",netsalary)

 else:

 netsalary=salary

 print("No taxalbe Amount")

 print("The net salary of "+e_name+" worked in " +c_name+ " is",salary)
```

39) Function computes the gross salary from basic salary.

```
def calcualte_gross_salary(basic_salary):
    hra = 0;
    da = 0;
    # salary is less than 2500, hra and da is calculated using this logic, otherwise else logic.
    if (basic_salary < 2500):
        hra = (basic_salary * 10) / 100;
        da = (basic_salary * 90) / 100;
    else:
        hra = 1000;
        da = (basic_salary * 95) / 100;

    return (basic_salary + hra + da);
if __name__ == "__main__":
    # Type casting from input string into float value.
    basic_salary = float(input("Enter basic salary: "));
    gross_salary = calcualte_gross_salary(basic_salary);
    print("Gross Salary is: %f" % gross_salary);
```

40) Write a program to create one array from another array.

```
#Initialize array
arr1 = [1, 2, 3, 4, 5];

#Create another array arr2 with size of arr1
arr2 = [None]* len(arr1);

#Copying all elements of one array into another
for i in range(0, len(arr1)):
    arr2[i] = arr1[i];

#displaying elements of array arr1
print("Elements of original array: ");
for i in range(0, len(arr1)):
   print(arr1[i]),

print();

#Displaying elements of array arr2
print("Elements of new array: ");
for i in range(0, len(arr2)):
   print(arr2[i]),
```

41) Write a program to sort the array elements using bubble sort technique.

```
def bubbleSort(arr):
    n = len(arr)

    # Traverse through all array elements
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]

# Array
arr = [64, 34, 25, 12, 22, 11, 90]

bubbleSort(arr)

print ("Sorted array is:")
for i in range(len(arr)):
    print ("%d" %arr[i]),
```

42) Write a program to pass a list to a function and display it.

```
def myFun(x):
   x[0] = 20


lst = [10, 11, 12, 13, 14, 15]
myFun(lst);
print(lst)
```

43) Write a program to create a list using range functions and perform append, update and delete elements operations in it.

```
list1=list(range(0,10))
print(list1)
#append
list1.append(12)
print(list1)
#update


#delete
list1.pop(2)
print(list1)
```

44) Create a sample list of 7 elements and implement the List methods mentioned: append, insert, copy, extend, count, remove, pop, sort, reverse and clear.

```
list1 =[1,22,33,30,5,2,99,9,]

print(list1)

#append

list1.append(12)

print(list1)

#Copy

print(list1.pop(5))

print(list1.insert(3, 22))

#copy

new_list=list1.copy()

print(new_list)

#extend

list1.extend(new_list)

print(list1)

#count

print(list1.count(1))

#remove
```

```
list1.remove(5)
print(list1)
#pop
list1.pop(10)
print(list1)
#sort
list1.sort()
print(list1)
#reverse
list1.reverse()
print(list1)
#clear
list1.clear()
print(list1)
```

45) Write a program to convert the elements of two lists into key-value pairs of a dictionary.

```
test_keys = ["Sam", "Devid", "Raj"]
test_values = [1, 2, 3]

print ("Original key list is : " + str(test_keys))
print ("Original value list is : " + str(test_values))

    res = {}

    for key in test_keys:

      for value in test_values:

        res[key] = value

        test_values.remove(value)

        break


    print ("Resultant dictionary is : " +  str(res))
```

46) Write a program to create a Student class with name, age and marks as data members. Also create a method named display() to view the student details. Create an object to Student class and call the method using the object.

```
class Student:

    def __init__(self,name,age,marks):

     self.ename=name

     self.eage=age

     self.emarks=marks


    def display(self):

       return (f"Student's Name is {self.ename} Age is {self.eage} Marks is {self.emarks}")
```

```
denish =Student("denish",19,87)

raj =Student("raj",20,88)

print(denish.display())
```

47) Write a program to create Student class with a constructor having more than one parameter.

```
class Student:

     def __init__(self,name,std,marks):

      self.ename=name

      self.estd=std

      self.emarks=marks


p1=Student("vedant",9,80)

print(p1.ename)
```

48) Write a program to demonstrate the use of instance and class/static variables.

```
class Demo:

   #Static Variable

   leaves=8
```

```
D1=Demo.leaves

print(D1)
```

49) Create a Bank class with two variables name and balance. Implement a constructor to initialize the variables. Also implement deposit and withdrawals using instance methods.

```
class Bank:

    def __init__(self,name,balance):

        self.ename=name

        self.ebalance=balance

    def display(self,deposit,withdrawls):

        return(f"Deposits{self.deposit}WithDrawls{self.withdrawls}")

b1=Bank("ABC",25000)

print(b1.ename)
```

50) Write a program to access the base class constructor from a sub class by using super() method and also without using super() method…….(This one is using super() method).

```
class A:

    no_of_elements1=10

    sp=3

    def __init__(self):

        self.var1="Class A"
```

```
class B(A):
    no_of_elements1="I am class B"
    def __init__(self):
        super().__init__()
        self.var2="Class B"
sanvi=A()
janvi=B()
print(janvi.var1)
```

51) Write a program to implement single inheritance in which two sub classes are derived from a single base class.

```
class A:
    p1=("Hello Class A")
class B(A):
    p2="cricket"
class C(A):
    p3="football"

obj1=B()
print(obj1.p1)
```

Second example of Single inheritance

```
class Person:
  def __init__(self, fname, lname):
    self.firstname = fname
    self.lastname = lname

  def printname(self):
    print(self.firstname, self.lastname)

class Student(Person):
  pass

x = Student("BCA", "SEM-5")
x.printname()
```

52) Write a program to implement multiple inheritance using two base classes.

```
class A:
    a1=5
class B:
    b1=("Hello class B")
```

```
class C(A,B):
   print("It's class C")
obj=C()
print(obj.b1)
print(obj.a1)
```

Second example of multiple inheritances

```
class Calculation1:
   def Summation(self,a,b):
      return a+b;
class Calculation2:
   def Multiplication(self,a,b):
      return a*b;
class Derived(Calculation1,Calculation2):
   def Divide(self,a,b):
      return a/b;
d = Derived()
print(d.Summation(10,20))
print(d.Multiplication(10,20))
print(d.Divide(10,20))
```

53) Write a program to show method overloading to find sum of two or three numbers.

```
class A:
    def func1(self):
        a=int(input("Enter A::"))
        b=int(input("Enter B::"))
        c=int(a+b)
        print("Addition::",c)
class B(A):
     def func(self):
        a=int(input("Enter A::"))
        b=int(input("Enter B::"))
        c=int(input("Enter C::"))
        d=int(a+b+c)
        print("Addition::",d)0
obj=B()
print(obj.func1())
```

54) Write a program to override the super class method in subclass.

```
class A:
    no_of_elements1=10
    number=3
    def __init__(self):
        self.var1="Class A"
class B(A):
    no_of_elements1="I am class B"
    def __init__(self):
        self.var1="Class B"
p1=A()
p2=B()
print(p2.number)
```