Naresh i Technologies, HYDERABAD

PYTHON HANDS ON EXAMPLES

1 Python Program to Calculate the Average of Numbers in a Given List

```
n=int(input("Enter the number of elements to be inserted: "))
a=[]
for i in range(0,n):
    elem=int(input("Enter element: "))
    a.append(elem)
avg=sum(a)/n
print("Average of elements in the list",round(avg,2))
```

2 Python Program to Exchange the Values of Two Numbers Without Using a Temporary Variable

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

3 Python Program to Read a Mumber n and Compute n+nn+nnn

```
n=int(input("Enter a number n: "))
temp=str(n)
t1=temp+temp
t2=temp+temp+temp
comp=n+int(t1)+int(t2)
print("The value is:",comp)
```

4 Python Program to Reverse a Given Number

```
n=int(input("Enter number: "))
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
print("Reverse of the number:",rev)
```

5 Python Program to Check Whether a Number is Positive or Negative

```
n=int(input("Enter number: "))
if(n>0):
    print("Number is positive")
else:
```

Prepared By: K.Subba Raju (19+Years of Experience)
Data Scientist & ML Expert

```
    print("Number is negative")
```

6 Python Program to Take in the Marks of 5 Subjects and Display the Grade

```
sub1=int(input("Enter marks of the first subject: "))
sub2=int(input("Enter marks of the second subject: "))
sub3=int(input("Enter marks of the third subject: "))
sub4=int(input("Enter marks of the fourth subject: "))
sub5=int(input("Enter marks of the fifth subject: "))
avg=(sub1+sub2+sub3+sub4+sub4)/5
if(avg>=90):
    print("Grade: A")
elif(avg>=80&avg<90):
    print("Grade: B")
elif(avg>=70&avg<80):
    print("Grade: C")
elif(avg>=60&avg<70):
    print("Grade: D")
else:
    print("Grade: F")
```

Python Program to Take in the Marks of 5 Subjects and Display result

```
sub1=int(input("Enter 1st subject Marks:"))
sub2=int(input("Enter 2nd subject Marks:"))
sub3=int(input("Enter 3rd subject Marks:"))
sub4=int(input("Enter 4th subject Marks:"))
sub5=int(input("Enter 5th subject Marks:"))
sub6=int(input("Enter 6th subject Marks:"))
if sub1>=40 and sub1<=100:
    print("Pass")
else :
    print("Failed")
if sub2>=40 and sub2<=100:
    print("Pass")
else :
    print("Failed")
if sub3>=40 and sub3<=100:
    print("Pass")
else :
    print("Failed")
if sub4>=40 and sub4<=100:
    print("Pass")
else :
    print("Failed")
if sub5>=40 and sub5<=100:
    print("Pass")
else :
```

```
    print("Failed")
if sub6>=40 and sub6<=100:
    print("Pass")
else :
    print("Failed")
```

Method-II

```
sub1=int(input("Enter the marks of Subject 1 :-"))
sub2=int(input("Enter the marks of Subject 2 :-"))
sub3=int(input("Enter the marks of Subject 3 :-"))
sub4=int(input("Enter the marks of Subject 4 :-"))
sub5=int(input("Enter the marks of Subject 5 :-"))
if sub1>100 or sub2>100 or sub3>100 or sub4>100 or sub5>100:
    print("INVALID MARKS YOU ENTERED")
else:
    if sub1<=49 or sub2<=49 or sub3<=49 or sub4<=49 or sub5<=49:
        print("YOU ARE FAIL")
    else:
        sum=sub1+sub2+sub3+sub4+sub5
        avg=sum/5
        print("Total Marks are:-",sum)
        print("Average is:-",avg)
        if(avg>=90 and avg<=100):
            print("YOU GOT A+ GRADE")
        elif(avg>=80 and avg<=90):
            print("YOU GOT A GRADE")
        elif(avg>=70 and avg<=80):
            print("YOU GOT B+ GRADE")
        elif(avg>=60 and avg<=70):
            print("YOU GOT B GRADE")
        elif(avg>=50 and avg<=60):
            print("YOU GOT C GRADE")
        else:
            print("YOU ARE FAIL")
```

7 Python Program to Print all Numbers in a Range Divisible by a Given Number

```
lower=int(input("Enter lower range limit:"))
upper=int(input("Enter upper range limit:"))
n=int(input("Enter the number to be divided by:"))
for i in range(lower,upper+1):
    if(i%n==0):
        print(i)
```

8 Python Program to Read Two Numbers and Print Their Quotient and Remainder

```
a=int(input("Enter the first number: "))
b=int(input("Enter the second number: "))
quotient=a//b
```

```
remainder=a%b
print("Quotient is:",quotient)
print("Remainder is:",remainder)
```

9 Python Program to Accept Three Digits and Print all Possible Combinations from the Digits

```
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
c=int(input("Enter third number:"))
d=[]
d.append(a)
d.append(b)
d.append(c)
for i in range(0,3):
    for j in range(0,3):
        for k in range(0,3):
            if(i!=j&j!=k&k!=i):
                print(d[i],d[j],d[k])
```

10 Python Program to Print Odd Numbers Within a Given Range

```
lower=int(input("Enter the lower limit for the range:"))
upper=int(input("Enter the upper limit for the range:"))
for i in range(lower,upper+1):
    if(i%2!=0):
        print(i)
```

11 Python Program to Find the Sum of Digits in a Number

```
n=int(input("Enter a number:"))
tot=0
while(n>0):
    dig=n%10
    tot=tot+dig
    n=n//10
print("The total sum of digits is:",tot)
```

12 Python Program to Find the Smallest Divisor of an Integer

```
n=int(input("Enter an integer:"))
a=[]
for i in range(2,n+1):
    if(n%i==0):
        a.append(i)
a.sort()
print("Smallest divisor is:",a[0])
```

13 Python Program to Count the Number of Digits in a Number

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

```
count=0
while(n>0):
    count=count+1
    n=n//10
print("The number of digits in the number are:",count)
```

14 Python Program to Check if a Number is a Palindrome

```
n=int(input("Enter number:"))
temp=n
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
if(temp==rev):
    print("The number is a palindrome!")
else:
    print("The number isn't a palindrome!")
```

15 Python Program to Print all Integers that Aren't Divisible by Either 2 or 3 and Lie between 1 and 50.

```
for i in range(0,51):
    if(i%2!=0&i%3!=0):
        print(i)
```

16 Python Program to Read a Number n And Print the Series "1+2+…..+n= "

```
n=int(input("Enter a number: "))
a=[]
for i in range(1,n+1):
    print(i,sep=" ",end=" ")
    if(i<n):
        print("+",sep=" ",end=" ")
    a.append(i)
print("=",sum(a))
print()
```

17 Python Program to Read a Number n and Print the Natural Numbers Summation Pattern

```
n=int(input("Enter a number: "))
for j in range(1,n+1):
    a=[]
    for i in range(1,j+1):
        print(i,sep=" ",end=" ")
        if(i<j):
            print("+",sep=" ",end=" ")
        a.append(i)
```

```
    print("=",sum(a))
print()
```

18 Python Program to Print an Identity Matrix

```
n=int(input("Enter a number: "))
for i in range(0,n):
    for j in range(0,n):
        if(i==j):
            print("1",sep=" ",end=" ")
        else:
            print("0",sep=" ",end=" ")
    print()
```

19 Python Program to Print an Inverted Star Pattern

```
n=int(input("Enter number of rows: "))
for i in range (n,0,-1):
    print((n-i) * ' ' + i * '*')
```

20 Python Program to Read Print Prime Numbers in a Range using Sieve of Eratosthenes

```
n=int(input("Enter upper limit of range: "))
sieve=set(range(2,n+1))
while sieve:
    prime=min(sieve)
    print(prime,end="\t")
    sieve-=set(range(prime,n+1,prime))
 print()
```

21 Python Program to Calculate the Average of Numbers in a Given List

```
n=int(input("Enter the number of elements to be inserted: "))
a=[]
for i in range(0,n):
    elem=int(input("Enter element: "))
    a.append(elem)
avg=sum(a)/n
print("Average of elements in the list",round(avg,2))
```

22 Python Program to Exchange the Values of Two Numbers Without Using a Temporary Variable

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

23 Python Program to Read a Mumber n and Compute n+nn+nnn

```
n=int(input("Enter a number n: "))
temp=str(n)
t1=temp+temp
t2=temp+temp+temp
comp=n+int(t1)+int(t2)
print("The value is:",comp)
```

24 Python Program to Reverse a Given Number

```
n=int(input("Enter number: "))
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
print("Reverse of the number:",rev)
```

25 Python Program to Check Whether a Number is Positive or Negative

```
n=int(input("Enter number: "))
if(n>0):
    print("Number is positive")
else:
    print("Number is negative")
```

26 Python Program to Take in the Marks of 5 Subjects and Display the Grade

```
sub1=int(input("Enter marks of the first subject: "))
sub2=int(input("Enter marks of the second subject: "))
sub3=int(input("Enter marks of the third subject: "))
sub4=int(input("Enter marks of the fourth subject: "))
sub5=int(input("Enter marks of the fifth subject: "))
avg=(sub1+sub2+sub3+sub4+sub4)/5
if(avg>=90):
    print("Grade: A")
elif(avg>=80&avg<90):
    print("Grade: B")
elif(avg>=70&avg<80):
    print("Grade: C")
elif(avg>=60&avg<70):
    print("Grade: D")
else:
    print("Grade: F")
```

27 Python Program to Print all Numbers in a Range Divisible by a Given Number

```
lower=int(input("Enter lower range limit:"))
upper=int(input("Enter upper range limit:"))
n=int(input("Enter the number to be divided by:"))
```

```
for i in range(lower,upper+1):
    if(i%n==0):
        print(i)
```

28 Python Program to Read Two Numbers and Print Their Quotient and Remainder

```
a=int(input("Enter the first number: "))
b=int(input("Enter the second number: "))
quotient=a//b
remainder=a%b
print("Quotient is:",quotient)
print("Remainder is:",remainder)
```

29 Python Program to Accept Three Digits and Print all Possible Combinations from the Digits

```
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
c=int(input("Enter third number:"))
d=[]
d.append(a)
d.append(b)
d.append(c)
for i in range(0,3):
    for j in range(0,3):
        for k in range(0,3):
            if(i!=j&j!=k&k!=i):
                print(d[i],d[j],d[k])
```

30 Python Program to Print Odd Numbers Within a Given Range

```
lower=int(input("Enter the lower limit for the range:"))
upper=int(input("Enter the upper limit for the range:"))
for i in range(lower,upper+1):
    if(i%2!=0):
        print(i)
```

31 Python Program to Find the Sum of Digits in a Number

```
n=int(input("Enter a number:"))
tot=0
while(n>0):
    dig=n%10
    tot=tot+dig
    n=n//10
print("The total sum of digits is:",tot)
```

32 Python Program to Find the Smallest Divisor of an Integer

```
n=int(input("Enter an integer:"))
```

```
a=[]
for i in range(2,n+1):
    if(n%i==0):
        a.append(i)
a.sort()
print("Smallest divisor is:",a[0])
```

33 Python Program to Count the Number of Digits in a Number

```
n=int(input("Enter number:"))
count=0
while(n>0):
    count=count+1
    n=n//10
print("The number of digits in the number are:",count)
```

34 Python Program to Check if a Number is a Palindrome

```
n=int(input("Enter number:"))
temp=n
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
if(temp==rev):
    print("The number is a palindrome!")
else:
    print("The number isn't a palindrome!")
```

35 Python Program to Print all Integers that Aren't Divisible by Either 2 or 3 and Lie between 1 and 50.

```
for i in range(0,51):
    if(i%2!=0&i%3!=0):
        print(i)
```

36 Python Program to Read a Number n And Print the Series "1+2+…..+n= "

```
n=int(input("Enter a number: "))
a=[]
for i in range(1,n+1):
    print(i,sep=" ",end=" ")
    if(i<n):
        print("+",sep=" ",end=" ")
    a.append(i)
print("=",sum(a))
print()
```

37 Python Program to Read a Number n and Print the Natural Numbers Summation Pattern

```
n=int(input("Enter a number: "))
for j in range(1,n+1):
    a=[]
    for i in range(1,j+1):
        print(i,sep=" ",end=" ")
        if(i<j):
            print("+",sep=" ",end=" ")
        a.append(i)
    print("=",sum(a))
print()
```

38 Python Program to Print an Identity Matrix

```
n=int(input("Enter a number: "))
for i in range(0,n):
    for j in range(0,n):
        if(i==j):
            print("1",sep=" ",end=" ")
        else:
            print("0",sep=" ",end=" ")
    print()
```

39 Python Program to Print an Inverted Star Pattern

ANS:

```
n=int(input("Enter number of rows: "))
for i in range (n,0,-1):
    print((n-i) * ' ' + i * '*')
```

40 Python Program to Read Print Prime Numbers in a Range using Sieve of Eratosthenes

```
n=int(input("Enter upper limit of range: "))
sieve=set(range(2,n+1))
while sieve:
    prime=min(sieve)
    print(prime,end="\t")
    sieve-=set(range(prime,n+1,prime))
 print()
```

LIST Examples:

41 Python Program to Find the Largest Number in a List

```
a=[]
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
    b=int(input("Enter element:"))
    a.append(b)
a.sort()
```

```
print("Largest element is:",a[n-1])
```

42 The program takes a list and prints the second largest number in the list.

```
a=[]
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
    b=int(input("Enter element:"))
    a.append(b)
a.sort()
print("Second largest element is:",a[n-2])
```

43 Python Program to Put Even and Odd elements in a List into Two Different Lists

```
a=[]
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
    b=int(input("Enter element:"))
    a.append(b)
even=[]
odd=[]
for j in a:
    if(j%2==0):
        even.append(j)
    else:
        odd.append(j)
print("The even list",even)
print("The odd list",odd)
```

44 Python Program to Merge Two Lists and Sort it

```
a=[]
c=[]
n1=int(input("Enter number of elements:"))
for i in range(1,n1+1):
    b=int(input("Enter element:"))
    a.append(b)
n2=int(input("Enter number of elements:"))
for i in range(1,n2+1):
    d=int(input("Enter element:"))
    c.append(d)
new=a+c
new.sort()
print("Sorted list is:",new)
```

45. Python Program to Sort the List According to the Second Element in Sublist

```
a=[['A',34],['B',21],['C',26]]
for i in range(0,len(a)):
    for j in range(0,len(a)-i-1):
```

```
        if(a[j][1]>a[j+1][1]):
            temp=a[j]
            a[j]=a[j+1]
            a[j+1]=temp
print(a)
```

46 Python Program to Find the Second Largest Number in a List Using Bubble Sort

```
a=[]
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
    b=int(input("Enter element:"))
    a.append(b)
for i in range(0,len(a)):
    for j in range(0,len(a)-i-1):
        if(a[j]>a[j+1]):
            temp=a[j]
            a[j]=a[j+1]
            a[j+1]=temp
print('Second largest number is:',a[n-2])
```

47 Python Program to Sort a List According to the Length of the Elements

```
a=[]
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
    b=input("Enter element:")
    a.append(b)
a.sort(key=len)
print(a)
```

48 Python Program to Find the Union of two Lists

```
a=[]
n= int(input("Enter the number of elements in list:"))
for x in range(0,n):
    element=int(input("Enter element" + str(x+1) + ":"))
    a.append(element)
b=[sum(a[0:x+1]) for x in range(0,len(a))]
print("The original list is: ",a)
print("The new list is: ",b)
```

49 Python Program to Find the Intersection of Two Lists

```
def intersection(a, b):
    return list(set(a) & set(b))
def main():
    alist=[]
    blist=[]
```

```
    n1=int(input("Enter number of elements for list1:"))
    n2=int(input("Enter number of elements for list2:"))
    print("For list1:")
    for x in range(0,n1):
        element=int(input("Enter element" + str(x+1) + ":"))
        alist.append(element)
    print("For list2:")
    for x in range(0,n2):
        element=int(input("Enter element" + str(x+1) + ":"))
        blist.append(element)
    print("The intersection is :")
    print(intersection(alist, blist))
main()
```

50 Python Program to Create a List of Tuples with the First Element as the Number and Second Element as the Square of the Number

```
l_range=int(input("Enter the lower range:"))
u_range=int(input("Enter the upper range:"))
a=[(x,x**2) for x in range(l_range,u_range+1)]
print(a)
```

51 Python Program to Find all Numbers in a Range which are Perfect Squares and Sum of all Digits in the Number is Less than 10

```
l=int(input("Enter lower range: "))
u=int(input("Enter upper range: "))
a=[]
a=[x for x in range(l,u+1) if (int(x**0.5))**2==x and sum(list(map(int,str(x))))<10]
print(a)
```

52 Python Program to Find the Cumulative Sum of a List where the ith Element is the Sum of the First i+1 Elements From The Original List

```
a=[]
n= int(input("Enter the number of elements in list:"))
for x in range(0,n):
    element=int(input("Enter element" + str(x+1) + ":"))
    a.append(element)
b=[sum(a[0:x+1]) for x in range(0,len(a))]
print("The original list is: ",a)
print("The new list is: ",b)
```

53 Python Program to Generate Random Numbers from 1 to 20 and Append Them to the List

```
import random
a=[]
n=int(input("Enter number of elements:"))
for j in range(n):
    a.append(random.randint(1,20))
print('Randomised list is: ',a)
```

54 Python program to Sort a List of Tuples in Increasing Order by the Last Element in Each Tuple

```
def last(n):
    return n[-1]
def sort(tuples):
    return sorted(tuples, key=last)
a=input("Enter a list of tuples:")
print("Sorted:")
print(sort(a))
```

55 Python Program to Swap the First and Last Value of a List

```
a=[]
n= int(input("Enter the number of elements in list:"))
for x in range(0,n):
    element=int(input("Enter element" + str(x+1) + ":"))
    a.append(element)
temp=a[0]
a[0]=a[n-1]
a[n-1]=temp
print("New list is:")
print(a)
```

56 Python Program to Remove the Duplicate Items from a List

```
a=[]
n= int(input("Enter the number of elements in list:"))
for x in range(0,n):
    element=int(input("Enter element" + str(x+1) + ":"))
    a.append(element)
b = set()
unique = []
for x in a:
    if x not in b:
        unique.append(x)
        b.add(x)
print("Non-duplicate items:")
print(unique)
```

57 Python Program to Read a List of Words and Return the Length of the Longest One

```
a=[]
n= int(input("Enter the number of elements in list:"))
for x in range(0,n):
    element=input("Enter element" + str(x+1) + ":")
    a.append(element)
max1=len(a[0])
temp=a[0]
for i in a:
    if(len(i)>max1):
```

```
        max1=len(i)
        temp=i
print("The word with the longest length is:")
print(temp)
```

58 Python Program to Remove the ith Occurrence of the Given Word in a List where Words can Repeat

```
a=[]
n= int(input("Enter the number of elements in list:"))
for x in range(0,n):
    element=input("Enter element" + str(x+1) + ":")
    a.append(element)
print(a)
c=[]
count=0
b=input("Enter word to remove: ")
n=int(input("Enter the occurrence to remove: "))
for i in a:
    if(i==b):
        count=count+1
        if(count!=n):
            c.append(i)
    else:
        c.append(i)
if(count==0):
    print("Item not found ")
else:
    print("The number of repetitions is: ",count)
    print("Updated list is: ",c)
    print("The distinct elements are: ",set(a))
```

59 Python Program to Remove All Tuples in a List of Tuples with the USN Outside the Given Range

```
y=[('a','12CS039'),('b','12CS320'),('c','12CS055'),('d','12CS100')]
low=int(input("Enter lower roll number (starting with 12CS):"))
up=int(input("Enter upper roll number (starting with 12CS):"))
l='12CS0'+str(low)
u='12CS'+str(up)
p=[x for x in y if x[1]>l and x[1]<u]
print(p)
```

60 To display odd number in the given numerical integer string

```
total=0
pylist=[1,2,3,4,5,6,7,8,9]
print('the odd numbers in the given list',pylist,'are:')
for i in range(0,9,1):
    if(pylist[i]%2!=0):
```

```
        print(pylist[i])
        total+=pylist[i]
print('\nSum of odd numbers of list is:',total)
```

61 To display even number in the given numerical integer string

```
total=0
pylist=[1,2,3,4,5,6,7,8,9]
print('the even numbers in the given list',pylist,'are:')
for i in range(0,9,1):
    if(pylist[i]%2==0):
        print(pylist[i])
        total+=pylist[i]
print('\nSum of even numbers of list is:',total)
```

62 To print first 'n' even numbers

```
n=int(input('enter the n value:\t'))
print('the even numbers between "1" and',n,'are:')
for x in range(1,n+1,1):
    if (x%2==0):
        print(x)
```

63 To find sum of first 'n' even numbers

```
total=0
n=int(input('enter the n value:\t'))
print('the even numbers between "1" and',n,'are:')
for x in range(1,n+1,1):
    if (x%2==0):
        total+=x
        print(x)
print('the sum of first',n,'even numbers are:',total)
```

64 To print even numbers between given range n1:n2

```
n1=int(input('enter the n1 value:\t'))
n2=int(input('enter the n2 value:\t'))
print('the even numbers between',n1,'and',n2,'are:')
for x in range(n1,n2+1,1):
    if (x%2==0):
        print(x)
```

65 To print first 'n' odd numbers

```
n=int(input('enter the n value:\t'))
print('the odd numbers between "1" and',n,'are:')
```

```
for x in range(1,n+1,1):
    if (x%2!=0):
        print(x)
```

66 To find sum of first 'n' odd numbers

```
total=0
n=int(input('enter the n value:\t'))
print('the odd numbers between "1" and',n,'are:')
for x in range(1,n+1,1):
    if (x%2!=0):
        total+=x
        print(x)
print('the sum of first',n,'odd numbers are:',total)
```

67 To print odd numbers between given range n1:n2

```
n1=int(input('enter the n1 value:\t'))
n2=int(input('enter the n2 value:\t'))
print('the odd numbers between',n1,'and',n2,'are:')
for x in range(n1,n2+1,1):
    if (x%2!=0):
        print(x)
```

68 To print first 'n' natural numbers

```
n=int(input('enter the n value:\t'))
print('the first',n,'natural numbers between "1" and',n,'are:')
for x in range(1,n+1,1):
    print(x)
```

69 To find sum of first 'n' natural numbers

```
total=0
n=int(input('enter the n value:\t'))
print('the natural numbers between "1" and',n,'are:')
for x in range(1,n+1,1):
    total+=x
    print(x)
print('the sum of first',n,'even numbers are:',total)
```

70 To print natural numbers between given range n1:n2

```
n1=int(input('enter the n1 value:\t'))
n2=int(input('enter the n2 value:\t'))
print('the natural numbers between',n1,'and',n2,'are:')
for x in range(n1,n2+1,1):
```

```
    print(x)
```

71 Print the multiplication table for given number with specific range[rows]

```
n1=int(input('enter the n1 value:\t'))
n2=int(input('how many rows:'))
for i in range(1,n2+1,1):
    print(n1,'X',i,'=',n1*i)
```

72 Print the multiplication table for given specific range between n1 and n2 with specific range[rows]

```
n1=int(input('enter the n1 value:\t'))
n3=int(input('how many rows:'))
for i in range(1,n1+1,1):
    for j in range(1,n3+1,1):
        print(i,'X',j,'=',j*i)
    print('-------------------------')
```

73 print the multiplication table for given specific range between n1 and n2 with specific range[rows]

```
n1=int(input('enter the n1 value:\t'))
n2=int(input('enter the n2 value:\t'))
n3=int(input('how many rows:'))
for i in range(n1,n2+1,1):
    for j in range(1,n3+1,1):
        print(i,'X',j,'=',j*i)
    print('-------------------------')
```

74 Unicode character of 'A' is 65

```
#print alphabets with unicodes from 'A' to 'Z'
for i in range(65,91,1):
    print('  ',chr(i),'==========>','unicode:',i)
```

75 Unicode character of 'a' is 97

```
#print alphabets with unicodes from 'a' to 'z'
for i in range(97,123,1):
    print('  ',chr(i),'==========>','unicode:',i)
```

76 find maximum number in given list

```
pylist=[1,2,3,5,8,9]
print('maximum number in given list ',pylist,' is:\t',max(pylist))
```

77 find minimum number in given list

```
pylist=[1,2,3,5,8,9]
print('minimum number in given list ',pylist,' is:\t',min(pylist))
```

78 find sum number in given list

```
pylist=[1,2,3,5,8,9]
print('sum of given list ',pylist,' is:\t',sum(pylist))
```

79 find len of in given list

```
pylist=[4,2,6]
s=sum(pylist)
c=len(pylist)
x=s/c
print('the average of number in given list ',pylist,' is:\t',x)
```

80 convert the negative integer numbers in the list into positive integer numbers

```
pylist=[1,2,3,4,-4,-6]
l=len(pylist)
for i in range(0,l,1):
    if (pylist[i]<0):
        pylist[i]*=-1
    print(pylist[i])
```

81 convert the positive integer numbers in the list into negative integer numbers

```
pylist=[1,2,3,4,-4,-6]
l=len(pylist)
for i in range(0,l,1):
    if (pylist[i]>0):
        pylist[i]*=-1
    print(pylist[i])
```

82 convert the odd integer numbers in the list into even integer numbers

```
pylist=[1,2,3,4,5,6,7,8]
l=len(pylist)
for i in range(0,l,1):
    if (pylist[i]%2!=0):
        pylist[i]+=1
    print(pylist[i])
```

83 convert the even integer numbers in the list into odd integer numbers

```
pylist=[1,2,3,4,5,6,88,6,7]
l=len(pylist)
for i in range(0,l,1):
    if (pylist[i]%2!=0):
        pylist[i]
    else:
        pylist[i]+=1
    print(pylist[i])
```

84 covert the all numbers in a list into same number list

```
n=int(input('enter what u want'))
pylist=[1,2,3,4,5,6,7]
l=len(pylist)
for i in range(0,l,1):
    if(pylist[i]==n):
        print(pylist[i])
    else:
        pylist[i]=n
        print(pylist[i])
```

85 Output of the following script

```
pylist=[1,2,3,4,[5,6,7,8,[9,10,11,12,[13,14,15,16]]]]
print(pylist)
print(pylist[4][4][4][3])
```

86 Output of the following script

```
#covert the all alphabets in a list into same alpha list
list1=['a','b','c','d','e']
c=str(input('enter your favourate letter:\t'))
a=ord(c)
l=len(list1)
for i in range(0,l,1):
    print(chr(a))
```

87 Output of the following script

```
#covert the all alphabets in a list into same string list
list1=['a','b','c','d','e']
c=str(input('enter your favourate letter:\t'))
l=len(list1)
for i in range(0,l,1):
    print(str(c))
```

Naresh i Technologies, HYDERABAD

Python Questions and Answers – Variable Names

1. Is Python case sensitive when dealing with identifiers?
a) yes
b) no
c) machine dependent
d) none of the mentioned

2. What is the maximum possible length of an identifier?
a) 31 characters
b) 63 characters
c) 79 characters
d) none of the mentioned

3. Which of the following is invalid?
a) _a = 1
b) __a = 1
c) __str__ = 1
d) none of the mentioned

4. Which of the following is an invalid variable?
a) my_string_1
b) 1st_string
c) foo
d) _

5. Why are local variable names beginning with an underscore discouraged?
a) they are used to indicate a private variables of a class
b) they confuse the interpreter
c) they are used to indicate global variables
d) they slow down execution

6. Which of the following is not a keyword?
a) eval
b) assert
c) nonlocal
d) pass

7. All keywords in Python are in
a) lower case
b) UPPER CASE
c) Capitalized
d) None of the mentioned

8. Which of the following is true for variable names in Python?
a) Unlimited length

Prepared By: K.Subba Raju (19+Years of Experience)
Data Scientist & ML Expert

b) All private members must have leading and trailing underscores
c) Underscore and ampersand are the only two special characters allowed
d) None of the mentioned

9. Which of the following is an invalid statement?

a) abc = 1,000,000
b) a b c = 1000 2000 3000
c) a,b,c = 1000, 2000, 3000
d) a_b_c = 1,000,000

10. Which of the following cannot be a variable?

a) __init__
b) in
c) it
d) on

Python Questions and Answers – Basic Operators

11. Which is the correct operator for power(x^y)?

a) X^y
b) X**y
c) X^^y
d) None of the mentioned

12. Which one of these is floor division?

a) /
b) //
c) %
d) None of the mentioned

13. What is the order of precedence in python?

i) Parentheses
ii) Exponential
iii) Division
iv) Multiplication
v) Addition
vi) Subtraction
a) i,ii,iii,iv,v,vi
b) ii,i,iii,iv,v,vi
c) ii,i,iv,iii,v,vi
d) i,ii,iii,iv,vi,v

14. What is answer of this expression, 22 % 3 is?

a) 7
b) 1
c) 0
d) 5

15. Mathematical operations can be performed on a string. State whether true or false.

a) True
b) False

16. Operators with the same precedence are evaluated in which manner?

a) Left to Right
b) Right to Left
c) Can't say
d) None of the mentioned

17. What is the output of this expression, 3*13?**

a) 27
b) 9
c) 3
d) 1

18. Which one of the following has the same precedence?

a) Addition and Subtraction
b) Multiplication and Division
c) Both a and b
d) None of the mentioned

19. The expression Int(x) implies that the variable x is converted to integer. State whether true or false.

a) True
b) False

20. Which one of the following has the highest precedence in the expression?

a) Exponential
b) Addition
c) Multiplication
d) Parentheses

21. Which of these in not a core data type?

a) Lists
b) Dictionary
c) Tuples
d) Class

22. Given a function that does not return any value, what value is thrown by default when executed in shell?

a) int
b) bool
c) void
d) None

23. Following set of commands are executed in shell, what will be the output?

1.>>>str="hello"
2.>>>str[:2]
3.>>>
a) he
b) lo
c) olleh
d) hello

24. Which of the following will run without errors?

a) round(45.8)
b) round(6352.898,2,5)
c) round()
d) round(7463.123,2,1)

25. What is the return type of function id ?

a) int
b) float
c) bool
d) dict

26. In python we do not specify types; it is directly interpreted by the compiler, so consider the following operation to be performed.

1.>>>x = 13 ? 2 objectives is to make sure x has a integer value, select all that apply (python 3.xx)
a) x = 13 // 2
b) x = int(13 / 2)
c) x = 13 % 2
d) All of the mentioned

27. What error occurs when you execute?

apple = mango
a) SyntaxError
b) NameError
c) ValueError
d) TypeError

28. Carefully observe the code and give the answer.

1.def example(a):
2. a = a + '2'
3. a = a*2
4. return a
5.>>>example("hello")

a) indentation Error
b) cannot perform mathematical operation on strings
c) hello2
d) hello2hello2

29. What datatype is the object below ?

L = [1, 23, 'hello', 1].
a) list

b) dictionary
c) array
d) tuple

30. In order to store values in terms of key and value we use what core datatype.

a) list
b) tuple
c) class
d) dictionary

31. Which of the following results in a SyntaxError ?

a) '"Once upon a time…", she said.'
b) "He said, 'Yes!'"
c) '3\'
d) "'That's okay"'

32. The following is displayed by a print function call:

1.tom
2.dick
3.harry
Select all of the function calls that result in this output
a) print("'tom
\ndick
\nharry"')
b) print("'tomdickharry"')
c) print('tom\ndick\nharry')
d) print('tom
dick
harry')

33. What is the average value of the code that is executed below ?

1.>>>grade1 = 80
2.>>>grade2 = 90
3.>>>average = (grade1 + grade2) / 2
a) 85
b) 85.1
c) 95
d) 95.1

34. Select all options that print

hello-how-are-you
a) print('hello', 'how', 'are', 'you')
b) print('hello', 'how', 'are', 'you' + '-' * 4)
c) print('hello-' + 'how-are-you')*
d) print('hello' + '-' + 'how' + '-' + 'are' + 'you')

35. What is the return value of trunc() ?

a) int
b) bool
c) float
d) None

36. What is the output of print 0.1 + 0.2 == 0.3?

a) True

b) False
c) Machine dependent
d) Error

37. Which of the following is not a complex number?
a) k = 2 + 3j
b) k = complex(2, 3)
c) k = 2 + 3l
d) k = 2 + 3J

38. What is the type of inf?
a) Boolean
b) Integer
c) Float
d) Complex

39. What does ~4 evaluate to?
a) -5
b) -4
c) -3
d) +3

40. What does ~~~~~~5 evaluate to?
a) +5
b) -11
c) +11
d) -5

41. Which of the following is incorrect?
a) x = 0b101
b) x = 0x4f5
c) x = 19023
d) x = 03964

42. What is the result of cmp(3, 1)?
a) 1
b) 0
c) True
d) False

43. Which of the following is incorrect?
a) float('inf')
b) float('nan')
c) float('56'+'78')
d) float('12+34')

44. What is the result of round(0.5) – round(-0.5)?
a) 1.0
b) 2.0
c) 0.0
d) None of the mentioned

45. What does 3 ^ 4 evaluate to?
a) 81 b) 12 c) 0.75 d) 7

46. The value of the expressions 4/(3*(2-1)) and 4/3*(2-1) is the same. State whether true or false.
a) True b) False

47. The value of the expression:
4 + 3 % 5
a) 4 b) 7 c) 2 d) 0
48. Evaluate the expression given below if A= 16 and B = 15.
A % B // A
a) 0.0 b) 0 c) 1.0 d) 1
49. Which of the following operators has its associativity from right to left?
a) + b) // c) % d) **
50. What is the value of x if: x = int(43.55+2/2)
a) 43 b) 44 c) 22 d) 23

GRAND REVIEW ON CORE PYTHON

1 Python Program to Reverse a Given Number

```
n=int(input("Enter number: "))
rev=0
while(n>0):
   dig=n%10
   rev=rev*10+dig
   n=n//10
print("Reverse of the number:",rev)
```

2 Python Program to Find the Sum of Digits in a Number

```
n=int(input("Enter a number:"))
tot=0
while(n>0):
   dig=n%10
   tot=tot+dig
   n=n//10
print("The total sum of digits is:",tot)
```

3 Python Program to Check if a Number is a Palindrome

```
n=int(input("Enter number:"))
temp=n
rev=0
while(n>0):
   dig=n%10
   rev=rev*10+dig
   n=n//10
if(temp==rev):
   print("The number is a palindrome!")
else:
   print("The number isn't a palindrome!")
```

4 Python Program to Print an Inverted Star Pattern

ANS:

```
n=int(input("Enter number of rows: "))
for i in range (n,0,-1):
```

```
   print((n-i) * ' ' + i * '*')
```

```
*****
 ****
  ***
   **
    *
```

5 Python Program to Swap the First and Last Value of a List

```
a=[]
n= int(input("Enter the number of elements in list:"))
for x in range(0,n):
   element=int(input("Enter element" + str(x+1) + ":"))
   a.append(element)
temp=a[0]
a[0]=a[n-1]
a[n-1]=temp
print("New list is:")
print(a)
```

6 Print the multiplication table for given number with specific range[rows]

```
n1=int(input('enter the n1 value:\t'))
n2=int(input('how many rows:'))
for i in range(1,n2+1,1):
   print(n1,'X',i,'=',n1*i)
```

7 Output of the following script

```
pylist=[1,2,3,4,[5,6,7,8,[9,10,11,12,[13,14,15,16]]]]
print(pylist)
print(pylist[4][4][4][3])#16
```

8 Output of the following script

```
pylist=[1,2,3,4,[5,6,7,8,[9,10,11,12,[13,14,15,16]]]]
print(pylist)
print(pylist[4][4][4][3])
```

9. Output of the following script

```
var1 = 'Hello NareshIT!'
var2 = "NareshITHYD"
print("var1[0]: ", var1[0])
print("var2[1:5]: ", var2[1:5])
```

10 What is the value of this expression: bin(10-2)+bin(12^4)

11 What is the output of print(['hello', 'morning'][bool(True)])

12 Count the Number of Vowels in a String

```
string=input("Enter string:")
vowels=0
for i in string:
    if(i=='a' or i=='e' or i=='i' or i=='o' or i=='u' or i=='A' or i=='E' or i=='I' or i=='O' or i=='U'):
        vowels=vowels+1
print("Number of vowels are:")
print(vowels)
```

13 Calculate the Number of Digits and Letters in a String

```
string=input("Enter string:")
count1=0
count2=0
for i in string:
    if(i.isdigit()):
        count1=count1+1
    count2=count2+1
print("The number of digits are:",count1)
print("The number of characters are:",count2)
```

14 Check a Number is a Prime Number or Not

```
a=int(input("Enter number: "))
k=0
for i in range(2,a//2+1):
   if(a%i==0):
      k=k+1
if(k<=0):
   print("Number is prime")
else:
   print("Number isn't prime")
```

15 Output of the following

```
for i in range(10):
   if i == 5:
      break
   else:
      print(i)
else:
   print("Here")
```

16 Output of the statement is

```
print('my_string'.isidentifier())
```

17 What is the output of the code shown below?

```
l=[2, 3, [4, 5]]
l2=l.copy()
l2[0]=88
print(l)
```

```
print(l2)
```

18 What is the output of the following code?

```
l1=[10, 20, 30]
l2=[-10, -20, -30]
l3=[x+y for x, y in zip(l1, l2)]
print(l3)
```

19 What is the output of the code shown below?

```
l1=[1,2,3]
l2=[4,5,6]
l3=[7,8,9]
for x, y, z in zip(l1, l2, l3):
        print(x, y, z)
```

20 What is the output of the code shown below?

```
a={}
print(a.fromkeys([1,2,3],"check"))
```

21 What is the output of the following code?

```
a={1,2,3}
b=a.add(4)
print(b)
```

22 What is the output of the following piece of code?

```
a=(2,3,1,5);a.sort();print(a)
```

23 What is the output of the following piece of code?

```
a=list((45,)*4)
print((45)*4)
print(a)
```

24 What is the output of the following piece of code?

```
print([[i+j for i in "abc"] for j in "def"])
[['ad', 'bd', 'cd'], ['ae', 'be', 'ce'], ['af', 'bf', 'cf']]
```

25 What is the output of the snippet of code shown below?

```
x=set('abcde')
y=set('xyzbd')
x.difference_update(y)
print(x)#{'a', 'c', 'e'}
print(y)#{'x', 'y', 'b', 'z', 'd'}
```

#To display odd number in the given numerical integer string
total=0

```
pylist=[1,2,3,4,5,6,7,8,9]
print('the odd numbers in the given list',pylist,'are:')
for i in range(0,9,1):
    if(pylist[i]%2!=0):
        print(pylist[i])
        total+=pylist[i]
print('\nSum of odd numbers of list is:',total)

#To display even number in the given numerical integer string
total=0
pylist=[1,2,3,4,5,6,7,8,9]
print('the even numbers in the given list',pylist,'are:')
for i in range(0,9,1):
    if(pylist[i]%2==0):
        print(pylist[i])
        total+=pylist[i]
print('\nSum of even numbers of list is:',total)

#To print first 'n' even numbers
n=int(input('enter the n value:\t'))
print('the even numbers between "1" and',n,'are:')
for x in range(1,n+1,1):
    if (x%2==0):
        print(x)

#To find sum of first 'n' even numbers
total=0
n=int(input('enter the n value:\t'))
print('the even numbers between "1" and',n,'are:')
for x in range(1,n+1,1):
    if (x%2==0):
        total+=x
        print(x)
print('the sum of first',n,'even numbers are:',total)

#To print even numbers between given range n1:n2
n1=int(input('enter the n1 value:\t'))
n2=int(input('enter the n2 value:\t'))
print('the even numbers between',n1,'and',n2,'are:')
for x in range(n1,n2+1,1):
    if (x%2==0):
        print(x)

#To print first 'n' odd numbers
n=int(input('enter the n value:\t'))
print('the odd numbers between "1" and',n,'are:')
```

```
for x in range(1,n+1,1):
    if (x%2!=0):
        print(x)

#To find sum of first 'n' odd numbers
total=0
n=int(input('enter the n value:\t'))
print('the odd numbers between "1" and',n,'are:')
for x in range(1,n+1,1):
    if (x%2!=0):
        total+=x
        print(x)
print('the sum of first',n,'odd numbers are:',total)

#To print odd numbers between given range n1:n2
n1=int(input('enter the n1 value:\t'))
n2=int(input('enter the n2 value:\t'))
print('the odd numbers between',n1,'and',n2,'are:')
for x in range(n1,n2+1,1):
    if (x%2!=0):
        print(x)

#To print first 'n' natural numbers
n=int(input('enter the n value:\t'))
print('the first',n,'natural numbers between "1" and',n,'are:')
for x in range(1,n+1,1):
    print(x)

#To find sum of first 'n' natural numbers
total=0
n=int(input('enter the n value:\t'))
print('the natural numbers between "1" and',n,'are:')
for x in range(1,n+1,1):
    total+=x
    print(x)
print('the sum of first',n,'even numbers are:',total)

#To print natural numbers between given range n1:n2
n1=int(input('enter the n1 value:\t'))
n2=int(input('enter the n2 value:\t'))
print('the natural numbers between',n1,'and',n2,'are:')
for x in range(n1,n2+1,1):
    print(x)

#Print the multiplication table for given number with specific range[rows]
n1=int(input('enter the n1 value:\t'))
```

```
n2=int(input('how many rows:'))
for i in range(1,n2+1,1):
    print(n1,'X',i,'=',n1*i)

#Print the multiplication table for given specific range between n1 and n2 with specific
range[rows]
n1=int(input('enter the n1 value:\t'))
n3=int(input('how many rows:'))
for i in range(1,n1+1,1):
    for j in range(1,n3+1,1):
        print(i,'X',j,'=',j*i)
    print('-------------------------')

#print the multiplication table for given specific range between n1 and n2 with specific
range[rows]
n1=int(input('enter the n1 value:\t'))
n2=int(input('enter the n2 value:\t'))
n3=int(input('how many rows:'))
for i in range(n1,n2+1,1):
    for j in range(1,n3+1,1):
        print(i,'X',j,'=',j*i)
    print('-------------------------')

#Unicode character of 'A' is 65
#print alphabets with unicodes from 'A' to 'Z'
for i in range(65,91,1):
    print('  ',chr(i),'=========>','unicode:',i)

#Unicode character of 'a' is 97
#print alphabets with unicodes from 'a' to 'z'
for i in range(97,123,1):
    print('  ',chr(i),'=========>','unicode:',i)

#find maximum number in given list
pylist=[1,2,3,5,8,9]
print('maximum number in given list ',pylist,' is:\t',max(pylist))

#find minimum number in given list
pylist=[1,2,3,5,8,9]
print('minimum number in given list ',pylist,' is:\t',min(pylist))

Example:
pylist=[1,2,3,5,8,9]
print('sum of given list ',pylist,' is:\t',sum(pylist))

Example:
```

```
pylist=[4,2,6]
s=sum(pylist)
c=len(pylist)
x=s/c
print('the average of number in given list ',pylist,' is:\t',x)
```

Example:

```
import matplotlib.pyplot as plt
x=[1,2,3,4,5]
plt.plot(x,'bo-')
plt.xlabel('time')
plt.ylabel('Y')
plt.title('wave with blue colour')
plt.show()
```

Example:

```
import matplotlib.pyplot as plt
x=[1,2,3,4,5]
plt.plot(x,'bo--')
plt.xlabel('time')
plt.ylabel('Y')
plt.title('wave with blue colour')
plt.show()
```

Example:

```
import matplotlib.pyplot as plt
x=[1,2,3,4,5]
plt.plot(x,'b-')
plt.plot(x,'ro')
plt.xlabel('time')
plt.ylabel('Y')
plt.title('wave with blue colour with red dots')
plt.show()
```

Example:

```
import matplotlib.pyplot as plt
x=[1,2,3,4,5]
plt.plot(x,'b--')
plt.plot(x,'ro')
plt.xlabel('time')
plt.ylabel('Y')
plt.title('dotted wave with blue colour with red dots')
plt.show()
```

```
#convert the negative integer numbers in the list into positive integer numbers
pylist=[1,2,3,4,-4,-6]
```

```
l=len(pylist)
for i in range(0,l,1):
    if (pylist[i]<0):
        pylist[i]*=-1
    print(pylist[i])


#convert the positive integer numbers in the list into negative integer numbers
pylist=[1,2,3,4,-4,-6]
l=len(pylist)
for i in range(0,l,1):
    if (pylist[i]>0):
        pylist[i]*=-1
    print(pylist[i])

#convert the odd integer numbers in the list into even integer numbers
pylist=[1,2,3,4,5,6,7,8]
l=len(pylist)
for i in range(0,l,1):
    if (pylist[i]%2!=0):
        pylist[i]+=1
    print(pylist[i])

#convert the even integer numbers in the list into odd integer numbers
pylist=[1,2,3,4,5,6,88,6,7]
l=len(pylist)
for i in range(0,l,1):
    if (pylist[i]%2!=0):
        pylist[i]
    else:
        pylist[i]+=1
    print(pylist[i])

#covert the all numbers in a list into same number list
n=int(input('enter what u want'))
pylist=[1,2,3,4,5,6,7]
l=len(pylist)
for i in range(0,l,1):
    if(pylist[i]==n):
        print(pylist[i])
    else:
        pylist[i]=n
        print(pylist[i])

Example:
pylist=[1,2,3,4,[5,6,7,8,[9,10,11,12,[13,14,15,16]]]]
```

```
print(pylist)
print(pylist[4][4][4][3])
```

Example:

```
#covert the all alphabets in a list into same alpha list
list1=['a','b','c','d','e']
c=str(input('enter your favourate letter:\t'))
a=ord(c)
l=len(list1)
for i in range(0,l,1):
    print(chr(a))
```

Example:

```
#covert the all alphabets in a list into same string list
list1=['a','b','c','d','e']
c=str(input('enter your favourate letter:\t'))
l=len(list1)
for i in range(0,l,1):
    print(str(c))
```

Example:

```
total=0
maximumtotal=0
counter=1
p=0
f=0
print('------------------------------------------------------------')
print('==================SMART PROGRESS
CARD=======================')
print('------------------------------------------------------------\n')
n=int(input('enter how many subjects:\t\t\t\t'))
print('------------------------------------------------------------\n')
counter
while counter<=n:
    sub=input('enter subject name:\t\t\t\t\t')
    m=int(input("Enter maximum marks:\t\t\t\t\t"))
    m1=int(input('Enter secured marks:\t\t\t\t\t'))
    while (m1>m or m1<0):
        print("==========Invalid typing==========")
        print('secured marks greaterthan amximum marks\n')
        m1=int(input("Enter secured marks:\t\t\t\t\t"))
    a=m1/m
    grade1=a*100
    if (grade1>50):
        p=p+1
    else:
```

```
        f=f+1
    if (grade1>100 or grade1<0):
        print('invalid condition')
print('================================================================')
    elif (grade1>=90 and grade1<=100):
        print('Grade:\t\t\t\t\t\t\tA+')
print('================================================================')
    elif (grade1>=80 and grade1<90):
        print('Grade:\t\t\t\t\t\t\tA')
print('================================================================')
    elif (grade1>=70 and grade1<80):
        print('Grade:\t\t\t\t\t\t\tB+')
print('================================================================')
    elif (grade1>=60 and grade1<70):
        print('Grade:\t\t\t\t\t\t\tB')
print('================================================================')
    elif (grade1>=50 and grade1<60):
        print('Grade:\t\t\t\t\t\t\tPASS')
print('================================================================')
    else:
        print('Grade:\t\t\t\t\t\t\tFAIL')
print('================================================================')
    total=total+m1
    maximumtotal=maximumtotal+m
    percent=total/maximumtotal
    grade=percent*100
    counter=counter+1
print('Summary:')
print('---------')
print('Pass subjects:\t\t\t\t\t',p)
print('Fail subjects:\t\t\t\t\t',f)
print('total secured marks:\t\t\t\t',total)
print('maximum marks:\t\t\t\t\t',maximumtotal)
if(f==0):
    print('total percentage of marks:\t\t\t',round(grade,2))
if (f==0 and grade>100 or grade<0):
    print('invalid condition')
print('================================================================')
elif (f==0 and grade>=90 and grade<=100):
    print('Overall Grade: \t\t\t\t\t A+')
print('================================================================')
elif (f==0 and grade>=80 and grade<90):
    print('Overall Grade:\t\t\t\t\t A')
print('================================================================')
elif (f==0 and grade>=70 and grade<80):
    print('Overall Grade:\t\t\t\t\t B+')
```

```
print('================================================================')
elif (f==0 and grade>=60 and grade<70):
    print('Overall Grade:\t\t\t\t\t\t B')
print('================================================================')
elif (f==0 and grade>=50 and grade<60):
    print('Overall Grade:\t\t\t\t\t\t PASS')
print('================================================================')
else:
    print('Overall Grade:\t\t\t\t\t\t FAIL')
print('================================================================')
print('================================================================')
print('Parent Signature:\t\t\t\t\t')
print('================================================================')
```

Q1. What is the difference between list and tuples in Python?

LIST	TUPLES
Lists are mutable i.e they can be edited.	Tuples are immutable (tuples are lists which can't be edited).
Lists are slower than tuples.	Tuples are faster than list.
Syntax: list_1 = [10, 'Chelsea', 20]	Syntax: tup_1 = (10, 'Chelsea' , 20)

Q2. What are the key features of Python?

- Python is an **interpreted** language. That means that, unlike languages like *C* and its variants, Python does not need to be compiled before it is run. Other interpreted languages include *PHP* and *Ruby*.
- Python is **dynamically typed**, this means that you don't need to state the types of variables when you declare them or anything like that. You can do things like x=111 and then x="I'm a string" without error
- Python is well suited to **object orientated programming** in that it allows the definition of classes along with composition and inheritance. Python does not have access specifiers (like C++'s public, private).
- In Python, **functions** are **first-class objects**. This means that they can be assigned to variables, returned from other functions and passed into functions. Classes are also first class objects
- **Writing Python code is quick** but running it is often slower than compiled languages. Fortunately Python allows the inclusion of C based extensions so bottlenecks can be optimized away and often are. The numpy package is a good example of this, it's really quite quick because a lot of the number crunching it does isn't actually done by Python

- Python finds **use in many spheres** – web applications, automation, scientific modeling, big data applications and many more. It's also often used as "glue" code to get other languages and components to play nice.

Q3. What type of language is python? Programming or scripting?

Ans: Python is capable of scripting, but in general sense, it is considered as a general-purpose programming language. To know more about Scripting, you can refer to the Python Scripting Tutorial.

Q4.How is Python an interpreted language?

Ans: An interpreted language is any programming language which is not in machine level code before runtime. Therefore, Python is an interpreted language.

Q5.What is pep 8?

Ans: PEP stands for **Python Enhancement Proposal.** It is a set of rules that specify how to format Python code for maximum readability.

Q6. How is memory managed in Python?

Ans:

1. Memory management in python is managed by ***Python private heap space***. All Python objects and data structures are located in a private heap. The programmer does not have access to this private heap. The python interpreter takes care of this instead.
2. The allocation of heap space for Python objects is done by Python's memory manager. The core API gives access to some tools for the programmer to code.
3. Python also has an inbuilt garbage collector, which recycles all the unused memory and so that it can be made available to the heap space.

Q7. What is namespace in Python?

Ans: A namespace is a naming system used to make sure that names are unique to avoid naming conflicts.

Q8. What is PYTHONPATH?

Ans: It is an environment variable which is used when a module is imported. Whenever a module is imported, PYTHONPATH is also looked up to check for the presence of the imported modules in various directories. The interpreter uses it to determine which module to load.

Q9. What are python modules? Name some commonly used built-in modules in Python?

Ans: Python modules are files containing Python code. This code can either be functions classes or variables. A Python module is a .py file containing executable code. Some of the commonly used built-in modules are:

- os
- sys
- math
- random
- data time
- JSON

Q10.What are local variables and global variables in Python?

Global Variables:

Variables declared outside a function or in global space are called global variables. These variables can be accessed by any function in the program.

Local Variables:
Any variable declared inside a function is known as a local variable. This variable is present in the local space and not in the global space.
Example:

```
1a=2
2def add():
3b=3
4c=a+b
5print(c)
6add()
```

Output: 5
When you try to access the local variable outside the function add(), it will throw an error.

Q11. Is python case sensitive?
Ans: Yes. Python is a case sensitive language.

Q12.What is type conversion in Python?
Ans: Type conversion refers to the conversion of one data type iinto another.
int() – converts any data type into integer type
float() – converts any data type into float type
ord() – converts characters into integer
hex() – converts integers to hexadecimal
oct() – converts integer to octal
tuple() – This function is used to convert to a tuple.
set() – This function returns the type after converting to set.
list() – This function is used to convert any data type to a list type.
dict() – This function is used to convert a tuple of order (key,value) into a dictionary.
str() – Used to convert integer into a string.
complex(real,imag) – This functionconverts real numbers to complex(real,imag) number.

Q13. How to install Python on Windows and set path variable?
Ans: To install Python on Windows, follow the below steps:

- Install python from this link: https://www.python.org/downloads/
- After this, install it on your PC. Look for the location where PYTHON has been installed on your PC using the following command on your command prompt: cmd python.
- Then go to advanced system settings and add a new variable and name it as PYTHON_NAME and paste the copied path.
- Look for the path variable, select its value and select 'edit'.
- Add a semicolon towards the end of the value if it's not present and then type %PYTHON_HOME%

Q14. Is indentation required in python?

Ans: Indentation is necessary for Python. It specifies a block of code. All code within loops, classes, functions, etc is specified within an indented block. It is usually done

using four space characters. If your code is not indented necessarily, it will not execute accurately and will throw errors as well.

Q15. What is the difference between Python Arrays and lists?

Ans: Arrays and lists, in Python, have the same way of storing data. But, arrays can hold only a single data type elements whereas lists can hold any data type elements.

Example:

```
1import array as arr
2My_Array=arr.array('i',[1,2,3,4
3])
My_list=[1,'abc',1.20]
4print(My_Array)
5print(My_list)
```

Output:

array(‘i’, [1, 2, 3, 4]) [1, ‘abc’, 1.2]

Q16. What are functions in Python?

Ans: A function is a block of code which is executed only when it is called.

Example:

```
1def Newfunc():
2print("Hi, Welcome to
3NareshIT")
Newfunc(); #calling the function
```

Output: Hi, Welcome to NareshIT

Q17.What is __init__?

Ans: __init__ is a method or constructor in Python. This method is automatically called to allocate memory when a new object/ instance of a class is created. All classes have the __init__ method.

Here is an example of how to use it.

```
1
2 class Employee:
3 def __init__(self, name,
4 age,salary):
5 self.name = name
6 self.age = age
7 self.salary = 20000
8 E1 = Employee("XYZ", 23, 20000)
9 # E1 is the instance of class
1 Employee.
0 #__init__ allocates memory for E1.
1 print(E1.name)
1 print(E1.age)
1 print(E1.salary)
```

Output:

XYZ
23
20000

Q18.What is a lambda function?

Ans: An anonymous function is known as a lambda function. This function can have any number of parameters but, can have just one statement.

Example:

```
a = lambda x,y :
x+y
print(a(5, 6))
```

Output: 11

Q19. What is self in Python?

Ans: Self is an instance or an object of a class. In Python, this is explicitly included as the first parameter. However, this is not the case in Java where it's optional. It helps to differentiate between the methods and attributes of a class with local variables.

The self-variable in the init method refers to the newly created object while in other methods, it refers to the object whose method was called.

Q20. How does break, continue and pass work?

Break

Allows loop termination when some condition is met and the control is transferred to the next statement.

Continue

Allows skipping some part of a loop when some specific condition is met and the control is transferred to the beginning of the loop

Pass

Used when you need some block of code syntactically, but you want to skip its execution. This is basically a null operation. Nothing happens when this is executed.

Q21. What does [::-1} do?

Ans: [::-1] is used to reverse the order of an array or a sequence.

Example:

```
import array as arr
My_Array=arr.array('i',[1,2,3,4,5])
My_Array[::-1]
```

Q22. How can you randomize the items of a list in place in Python?

```
from random import shuffle
x = ['Keep', 'The', 'Blue', 'Flag', 'Flying', 'High']
shuffle(x)
print(x)
```

Q23. How do you write comments in python?

Ans: Comments in Python start with a # character. However, alternatively at times, commenting is done using docstrings(strings enclosed within triple quotes).

Example:

```
#Comments in Python start like this

print("Comments in Python start with a #")
```

Q24. What is pickling and unpickling?

Ans: Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

Q25. What are the generators in python?

Ans: Functions that return an iterable set of items are called generators.

Q26. How will you capitalize the first letter of string?

Ans: In Python, the capitalize() method capitalizes the first letter of a string. If the string already consists of a capital letter at the beginning, then, it returns the original string.

Q27. How will you convert a string to all lowercase?

```
stg='ABCD'

print(stg.lower())
```

Q28. What does this mean: *args, **kwargs? And why would we use it?

Ans: We use *args when we aren't sure how many arguments are going to be passed to a function, or if we want to pass a stored list or tuple of arguments to a function. **kwargs is used when we don't know how many keyword arguments will be passed to a function, or it can be used to pass the values of a dictionary as keyword arguments. The identifiers args and kwargs are a convention, you could also use *bob and **billy but that would not be wise.

Q29. How is Multithreading achieved in Python?

Python has a multi-threading package but if you want to multi-thread to speed your code up, then it's usually not a good idea to use it.

Python has a construct called the Global Interpreter Lock (GIL). The GIL makes sure that only one of your 'threads' can execute at any one time. A thread acquires the GIL, does a little work, then passes the GIL onto the next thread.

This happens very quickly so to the human eye it may seem like your threads are executing in parallel, but they are really just taking turns using the same CPU core.

All this GIL passing adds overhead to execution. This means that if you want to make your code run faster then using the threading package often isn't a good idea.

Q30. What is monkey patching in Python?

Ans: In Python, the term monkey patch only refers to dynamic modifications of a class or module at run-time.

```
# m.py

class MyClass:

def f(self):

print "f()"
```

We can then run the monkey-patch testing like this:

```
import m

def monkey_f(self):

print "monkey_f()"


m.MyClass.f = monkey_f

obj = m.MyClass()

obj.f()
```