Section 1 :- Theory

Q1. What is the difference between list and tuples?

--List are mutable, whereas tuples are immutable data objects. Both are sequential objects. They differ by the way of representation. Lists use [ ] i.e., square brackets, Tuples use ( ) parenthesis.

Q2. What are the key features of Python?

--Python provides lots of features, like:

- Open Source programming
- General Purpose programming Language
- Dynamic programming Language
- Object Oriented programming
- Interpreted programming language
- Cross-platform language /Platform Independent
- Provides GUI programming support
- Integrated programming

Q3. What is the difference between deep and shallow copy?

-- A Deep Copy copies all fields and makes copies of dynamically allocated memory pointed to by the fields. It also copies the object along with the objects to which it refers.

-- A Shallow copy is just a bitwise copy of an object.

--Mutating an object that was deep-copied doesn't change the existing object. In other words, deep copying gives the option of both the objects (parent and the copy) being fully independent.

Example:

Shallow Copy:

```
x_list=[1, 2, 3, 4, 5]
y_list = x_list
```

#ShallowCopy :Changing any elements in x_list, will modify that in y_list

Hard Copy:

```
        import copy
        z_list=copy.deepcopy(x_list)
```

#Change in the x_list elements doesnot reflect on the z_list elements.

Q4. How can the ternary operators be used in python?

--Ternary Operators, also called as Conditional operators are used to evaluate whether the condition is True or False, and thereby to have the flow of control that specifies when its executed.

.

Q5. How is memory managed in Python?

--The memory management involves a 'Private heap' that stores all the Objects and data structures. The management of 'Private heap' is ensured internally by the 'Python Memory Manager'.

--At the lowest level, it ensures that it has enough storage space for data in the private heap, by interacting with the raw memory allocator. On top of the raw memory allocator, several object specific allocators operate on the same heap and implement the policies that are particular to the defined objects.

--The management of the Heap is performed by the interpreter itself and the user has no control over it.

Q6. What is the usage of help() and dir() function in Python?

--dir(): This function can be used to list out the available functions in the specified module or packages.

--help(): Function can be used to display the documentation, basic syntax to follow, that's associated with the function or module that's needed.

Q7. Whenever Python exits, why isn't all the memory de-allocated?

--Objects referenced form the modules are not always de-allocated, when Python exits. This happens when there are circular references. Also, Clibrary allocates certain memory, that are impossible to erase by the Python. However, Python tries to erase through its own efficient clean-up mechanism and try to destroy every object that doesn't have any global references.

Q8. What are negative indexes and why are they used?

--Negative indexes are the indexes given to the sequential objects to refer from the end towards the beginning of the sequence, if indexing is allowed (unlike sets)

--They are used based on the need to access the elements from the end of the sequence, or in other words, to reverse the order of indexing.

Q9. What is the difference between range & xrange?

--range() :generates a static list of integers at runtime, with the specified minium, maximum and step, given as arguments.

--xrange(): doesn't generate a static list like range(). It only generates the values as needed, through a technique called 'yielding'. xrange() is only available in Python 2.x. In Python 3.x, range() does the what xrange() can do in Python2.x.

Q10. Which one of these is floor division?

--Floor Division in python : //

```
example: [In}: 5//2
               [Out]: 2
```

Section 2 :- Programming Analyzation

----

1) Find the output of the code:

```
sum=0
for i in range(12,2,-2):
        sum+=i
print sum
```

[Out]: 40 #12+10+8+6+4

2) Find the output of the following code:

```
n=50
i=5
s=0
while i<n:
        s+=i
        i+=10
print “i=”,i
print “sum=”,s
```

[Out]: i: 55
        sum=125

3) Find the output of the following code:

```
List=[1, 6, 8, 4, 5]
print List[-4:]
```

[Out]: [6, 8, 4, 5]

4) How many times are the following loops executed?

```
i=100
while(i<=200):
        print i
        i+=20
```

[Out]: 6 times. Starts with 100 and keeps appending 20 to i, until 200,
        100, 120, 140, 160, 180, 200.

5) Find the output of the following code:

```
L= [100, 200, 300, 400, 500]
L1= L[2:4]
print L1
L2= L[1:5]
print L2
L2.extend(L1)
print L2
```

```
[Out]: [300, 400] #L1
       [200, 300, 400, 500] #L2
       [200,300,400, 500, 300, 400] #L2.extend(L1)
```

6) Predict the behavior of the code:

```
List=list(“String”)
print List
```

```
[Out]: [‘S’, ‘t’, ‘r’, ‘i’, ‘n’, ‘g’] #Converts a string object to list object.
```

Section 3 :- Problem solving

Python Program to Convert Kilometers to Miles

```
#To Convert Kilometers to miles:
def main():
     km= int(input(“Enter the Kilometers:”))
     miles= km*(0.621371)
     print “The miles are:”, miles

if(__name__==”__main__”):
     main()
```

Python Program to Convert Celsius To Fahrenheit

```
#To convert Celsius to Fahrenheit:

def main():
     celsius=int(input("Enter the temperature in
          Celsius: "))
     fahrenheit= (celsius *(1.8))+32
     print "Temperature in Fahrenheit:
          ",Fahrenheit

 if (__name__=="__main__"):
      main()
```

Python Program to Check Prime Number

```
#To check if the given number is Prime:

def main():
     number=int(input("Enter the number:"))
     for i in range(2,number):
        if number%i==0:
             print "The number is NOT a Prime
Number."
             break
    else:
         print "The number is a Prime Number."

 if (__name__=="__main__"):
      main()
```

Python Program to Print all Prime Numbers in an Interval

```
#To print all the Prime numbers in a range.
def main():
     interval=int(input("Enter the range of
                         PrimeNumbers needed: "))
     print "Prime Numbers are: ",
     for num in range(2,interval):
          for i in range(2,num):
               if num%i==0:
                    break
          else:
               print num,

if (__name__=="__main__"):
     main()
```

Python Program to Find the Factorial of a Number

```
#To get the Factorial of a number
def main():
     num=int(input(“Enter the number: “))
     prod=1
     print “The Factorial is: “,
     for i in range(2,num+1):
          prod*=i
     print prod

def (__name__==”__main__”):
     main()
```

Python Program to Display the multiplication Table

```
#To print out the Multiplication Table
def main():
     num=int(input(“Which multiplication Table: “))
     for i in range(1,21):
          print num,”X”,i,”=”,(num*i)

if (__name__==”__main__”):
     main()
```

Python Program to Print the Fibonacci sequence

```
#Fibonacci Sequence:
def main():
     num=int(input(“Enter the range for the Fibonacci
                    Sequence:”))
     first=0
     second=1
     third=first+second
     print first,second,
     while third<=num:
          print third,
          first=second
          second=third
          third=first+second

if (__name__==”__main__”):
     main()
```

Python Program to Check Armstrong Number

```
#To check for Armstrong Number
def main():
     num=int(input(“Enter the number to check
                    w/Armstrong number: “))
     n=num
     sum=0
     L=[]
     while n>0:
          L.append(n%10)
          n=n//10
     for i in range(len(L)):
          sum+=(L[i])**3
     if sum==num:
          print “The number is an Armstrong number”
     else:
          print “The number is NOT an Armstrong number”

if (__name__==”__main__”):
     main()
```

Python Program to Find HCF or GCD

```
#Find the HCF or GCD
def main():
     a=int(input(“Enter the first number: “))
     b=int(input(“Enter the second number: “))
     small=min(a,b)
     big=max(a,b)
     gcd=0
     print “The HCF or GCD is: “,
     for i in range(1,small+1):
          if (small%i==0) and (big%i==0):
               gcd=i
     print gcd

if(__name__==”__main__”):
     main()
```

Python Program to Find LCM

```
# Find the LCM
def main():
     a=int(input(“Enter the first number: “))
     b=int(input(“Enter the second number: “))
     big=max(a,b)
     lcm=0
     print “The LCM is: “,
     while True:
          if big%a==0 and big%b==0:
               lcm=big
               break
          big+=1
     print lcm

if(__name__==”__main__”):
     main()
```

Python Program to Find Factors of Number

```
#To get the Factors of a number
def main():
     num=int(input(“Enter the number: “))
     print “The Factors are: “,
     for i in range(1,num+1):
          if number%i==0:
               print i,

def (__name__==”__main__”):
     main()
```