Conditional and Looping Constructs

Very Short Ans type Quess [1 mark each]

Ques 1:

Why do we use 'break' statement ?

Ans:

The 'break' statement can be used to terminate the loop.

Ques 2:

What gets printed with the following code ?

```
x = True
y = False
z = False
if x or y and z :
print "yes"
else:
print "no
```

Ans:

Yes

Ques 3:

What gets printed with the following code ?

```
x = True
y = False
z = False
if not x or y :
print 1
elif not x or not y and z:
print 2
elif not x or y or not y and x:
print 3
else:
```

print 4
Ans: 3

Ques 4:
What gets printed with given code ?
f = None
for i in range (5):
with open(“data.txt”, “w”) as f:
if i > 2:
break
print f.closed
Ans:
True

Ques 5:
Which numbers are printed?
for i in range(2):
print i
for i in range(4,6):
print i
Ans:
0,1,4, 5

Ques 6:
What gets printed?
import re
sum = 0
pattern = ‘back’
if re.match(pattern, ‘backup.txt’):
sum + = 1
if re.match(pattern, ‘text.back’):

sum + = 2
if re.search(pattern, ‘backup.txt’):
sum + = 4
if re.search(pattern, ‘text.back’):
sum + = 8
print sum
Ans:
13

Ques 7:
Write the syntax of an ’if statement1 in Python programming language
Ans:
if expression :
statement(s)

Ques 8:
Write the syntax of an if…..else statement in Python programming language
Ans:
if expression:
statement(s)
else:
statement(s)

Ques 9:
Write the output
#!/usr/bin/python
var =100
if(var==100):
print “Value of expression is 100”
print “Good bye!
Ans:

Value of expression is 100
Good bye

Ques 10:
Define while loops.
Ans:
A while loop statement in Python programming language repeatedly executes a target statement as long as a given condition is true.

Ques 11:
Write the syntax of a while loop.
Ans:
The syntax of a while loop in Python programming language is :
while expression :
statement(s)

Ques 12:
What happened when the condition of while loop becomes false ?
Ans:
When the condition becomes false, program control passes to the line immediately following the loop.

Ques 13:
Write the syntax of a for loop.
Ans:
The syntax of a for loop look is as follows:
for iterating_var in sequence :
statements(s)

Ques 14: What do you mean by “continue statement” ?
Ans: It causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.
Ques 15: What do you mean by “pass statement” ?
Ans: The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.
Ques 16:
Write the syntax of “break” statement.
Ans:
The syntax for a break statement in Python is as follows : break

Ques 17:
Does Python support “switch” statements?
Ans:
No, Python does not currently support switch or case statements as in other languages.

Short Ans type Quess [2 mark each]

Ques 1:
What is a statement ? What is the significance of an empty statement ?
Ans:
A statement is an instruction given to the computer to perform any kind of action.
An empty statement is useful in situations where the code requires a statement but does not require logic. To fill these two requirements simultaneously, empty statement is used. Python offers ‘pass’ statement as an empty statement.

Ques 2:
What is the difference between determinable loop and non-determinable loop ?
Ans:
The ‘for loop’ can be labelled as ‘determinable
Ans. Value of expression is 100 Good bye!
loop’ as number of its iterations can be determined before-hand as the size of the sequence, it is operating upon.
The ‘while loop’ can be ‘non-determined loop’ as its number of iterations cannot be determined before-hand. Its iterations depend upon the result of a test-condition, which cannot be determined before-hand.

Ques 3: What are the two types of else clause in Python ? Ans. The two types of Python else clauses are :
Ans:
(i) else in an if statement
(ii) else in a loop statement
The else clause of an if statement is executed when the condition of the if statement results into false.
The else clause of a loop is executed when the loop is terminating normally i.e. when its test-condition has gone false for a while loop or when the for loop has executed the last value in sequence.

Ques 4: Explain nested if…. else.
Ans:
There may be a situation when you want to check for another condition after a condition resolves to true. In such a situation, you can use the nested if construct. In a nested if construct, you can have an if…elif…else construct inside another if…elif… else construct

Ques 5:

Write the syntax of a for loop and also given an example.

Ans:

The syntax of a for loop looks as follows : for iterating_var in sequence: statements(s)

Example : for i in range (4):

print i

output : 0 1 2 3

Ques 6:

What do you mean by “for loop” ?

Ans:

A for loop is à Python statement which repeats a group of statements a specified number of times. You can use any object (such as strings, arrays, lists, tuples, diet and so on) in a for loop in Python.

Ques 7:

Explain If…else statements

Ans:

If…else statements

An else statement can be combined with an if statement. An else statement contains the block of code that executes if the conditional expression in the if statement resolves to 0 or a false value.

The else statement is an optional statement and there could be at most only one else statement following if

Syntax:

The syntax of the if…else statement is :

if expression :

statement(s)

else:

statement(s)

Ques 8:

What do you mean by decision making ?

Ans:

Structures require that the programmer specify one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be- true, and optionally, other statements to be executed if the condition is determined to be false.

Ques 9:

Create a function addNumbers(x) that takes a number as an argument and adds all the integers between 1 and the number (inclusive) and returns the total number.

Ans:

```
def add Numbers (num):
total = 0
i = 1
while i< = num:
total + = i
i+ = 1
return total
```

Ques 10:

Create a function addNumbers(start, end) that adds all the integers between the start and end value (inclusive) and returns the total sum.

Ans:

```
def addNumbers(start, end):
total = 0
i = start
while start < = end:
total + = start
```

```
start + = 1
return total
```

Ques 11:

Create a function countPages(x) that takes the number of pages of a book as an argument and counts the number of times the digit '1' appears in the page number.

Ans:

```
def countPages(num):
total = 0
i = 1
while i<=num:
page_no = str(i)
total + = page_no.count('1')
i+ = l
return total
```

Ques 13:

Create a function factorial(x) that takes an integer and returns the product of all the positive integers less than or equal to n.

Ans:

```
def factorial(num):
product =1
i = num
while i > 0:
product *=i
i-=l
return product
```

Ques 14:

Create a function doubleFactorial(n) that takes an odd integer and returns the product of all odd values upto the value n (where n=2k-l).

Ans:

```
def doubleFactorial(num):
product = 1
i = 0
k = 0
while k <num:
k = 2*i+l
product * = k
i += 1
return product
```

Ques 15:

Create a function that computes the approximation of pi, based on the number of iterations specified, pi can be computed by 4*(l-l/3+l/5-l/7+l/9-…).

Ans:

```
def piApprox(num):
i = 1
pi = 0
while i<=num:
#set 'while' termination condition
pi +=((4/float(2*i-l)n-l)**(i+l))
#compute the ith term of the series
i+=l    # update i
return pi
```

Ques 16:

What gets printed?

```
country_counter = {}
def addone(country):
if country in country_counter:
country_counter[country] + = 1
else:
country_counter[country] = 1
addone('China')
addone('Japan')
addone('china')
print len(country_counter)
```

Ans:

3

Ques 17:

What gets printed?

```
namesl = ['Amir', 'Barry', 'Chales', 'Dao']
if 'amir' in namesl:
print 1
else:
print 2
```

Ans:

2

Ques 18:

What gets printed?

```
x = 0
y = l
a = cmp(x,y)
if a < x:
```

```
print “a”
elif a = = x:
print “b”
else:
print “c”
```

Ans:

a

Ques 19:

What gets printed ?

```
x = 1
y = “2”
z = 3
sum = 0
for i in (x,y,z):
if isinstance(i, int):
sum + = i
print sum
```

Ans:

4

Long Ans type Quess [4 mark each]

Ques 1:

Write an example to illustrates the combination of an else statement with a for statement

Ans:

The following example illustrates the combination of an else statement with a for statement that searches for prime numbers from 10 through 20.

```
# !/usr/bin/python
for num in range(10,20):
#to iterate between 10 to 20
for i in range(2,num):
#to iterate on the factors of the number
if num%i == 0:
#to determine the first factor
j=num/i
#to calculate the second factor
print '%d equals %d * %d' % (num,i,j)
break
#to move to the next number, the
#first for
else:
# else part of the loop
print num, 'is a prime number'
```

When the above code is executed, it produces following result :

```
10 equals 2*5
11 is a prime number
12 equals 2*6
13 is a prime number
14 equals 2*7
```

15 equals 3*5
16 equals 2*8
17 is a prime number
18 equals 2*9
19 is a prime number

Ques 3:

Write a program to explain “for loop”

Ans:

```
#!/usr/bin/python
for letter in ‘Python’:    # First Example
print ‘Current Letter :’, letter
fruits = [‘banana’, ‘apple’, ‘mango’]
for fruit in fruits:    # Second Example
print ‘Current fruit :’, fruit
print “Good bye!”
```

When the above code is executed, it produces following result:

```
Current Letter : P
Current Letter : y
Current Letter : t
Current Letter : h
Current Letter : o
Current Letter : n
Current fruit : banana
Current fruit : apple
Current fruit : mango
Good bye!
```

Ques 4:

Create a function that takes in a positive integer and return a list of prime numbers. A prime number is only divisible by 1 and itself.

Ans:

```
def primeNumbers(num):
primes = [ ]
i = 2
# iterates through range from 2 to
num(inclusive)
while i< =num:
# add ‘while’ condition
k = 2
is Prime = True
# check if prime number
while k<i: # add’while’condition
if i%k==0:
isPrime = False
k+ = l   # update k
if isPrime = =True:
primes.append(i)
i+ = l # update i
return primes
```

Ques 5:

Create a function that takes in a positive number and return 2 integers such that the number is between the squares of the 2 integers. It returns the same integer twice if the number is a square of an integer.

Ans:

```
import math
def sqApprox(num):
i = 0
```

```
minsq = 0 # set lower bound
maxsq = math.ceil(num'l'num)
# set upper bound
while i< maxsq :
# set 'while' termination condition
if i*i< =num and i>minsq:
# complete inequality condition
minsq = i
if i*i> =num and i<maxsq:
# complete inequality condition
maxsq = i
i+=l
# update i so that 'while' will terminate
return (minsq, maxsq)
```

Ques 6:

Write a function estimatePi( ) to estimate and return the value of pi based on the formula found by an Indian Mathematician Srinivasa Ramanujan. It should use a while loop to compute the terms of the summation until the last term is smaller than le-15. The formula for estimating pi is given below :

where k! is the factorial of k.

Ans:

```
import math
def factorial(n):
if n = = 0:
return 1
else:
return n * factorial(n-l)
```

```
def series_term(k):
a=factorial(4*k)
b=(1103+26390*k)
c=factoriall(k)
d=c**4
e=396**(4*k)
return float(a'fb)/(d'l'e)

def estimatePi():
k=0
final=0
while True:
term = series_term(k)
final + = term
k+=l
if term < 1.0e-15:
break
f=2*math.sqrt(2)/9801
pi = 1.0/(final * f)
return pi
```

Ques 7:

Given a positive integer, write a function that computes the prime factors that can be multplied together to get back the same integer.

Ans:

```
def primeFactorization(num):
factors = [ ]
lastresult = num
# 1 is a special case
if num == 1:
return [ ]
```

```
while True:
if lastresult == 1:
break
c = 2
while True:
if lastresult % c = = 0:
break
c += 1
factors.append(c)
lastresult/= c
return factors
```

Ques 8:

The smallest common multiple of two or more numbers is called the lowest common multiple (LCM). Given a list of integers, find the lowest common multiple.

Ans:

```
def LCM(nums):
nums.sort()
biggest=nums[-l]
multiplier=1
while sum([(multiplier'|,biggest)%num for num in nums])!=0:
multiplier + = 1
return biggest*multiplier
```

Ques 9:

Write a program to explain "if statement"

Ans:

```
# !/usr/bin/python
var1=100
if var1:
```

```
print “1-Got a true expression Value”
print var1
else:
print ” 1-Got a false expression value”
print var1 var2=0
if var2 :
print “2-Got a true expression value”
print var2
else :
print “2-Got a false expression value”
print var2
print “Good bye!”
```

When the above code is executed, it produces following result:

```
1- Got a true expression value 100
2- Got a false expression value 0
Good bye!
```

Ques 10:

Explain “elif statement” with example.

Ans:

elif statement : The elif statement allows you to check multiple expressions for truth value and execute a block of code as soon as one of the conditions evaluates to true. Like the else, the elif statement is optional. However, unlike else , for which there can be at most one statement, there can be an arbitrary number of elif statements following an if

The syntax of the if…elif statement is

```
If expression1 :
statement (s)
elif expression2 :
statement(s)
```

elif expression3 :
statement(s)
else:
statement(s)

Example

```
# !/usr/bin/py thon
var=100
if var = = 200 :
print “1-Got a true expression value”
print var
elif var==150:
print “2-Got a true expression value”
print var2
elif var ==100:
print “3-Got a true expression value”
print var
else:
print “4-Got a false expression value”
print var
print “Good bye!”
```

when the above code is executed, it produces the following output :

```
3- Got a true expression value
100
Good bye !
```

Ques 11:

Explain nested if—–else statement

Ans:

There may be a situation when you want to check for another condition after a condition resolves to true. In such a situation, you can use the nested if construct. In a nested if construct, you can have

an if…elif…else construct inside another if…elif… else construct.

Syntax:

The syntax of the nested if…elif…else construct may be:

```
if expression1:
statement(s)
if expression2:
statement(s)
elif expression3:
statement(s)
else
statement(s)
elif expression4: statement(s) else:
statement(s)
```

Example:

```
# !/usr/bin/python if var < 200:
print “Expression value is less than 200”
if var==150:
print “=150”
elif var = =100:
print “=100”
elif var ==50:
print “=50”
elif var < 50 :
print “Expression value is less than 50”
else :
print “Could not find true expression”
```

Ques 12:

Explain while loop in Python programming language.

Ans:

The syntax of a while loop in Python programming language is

while expression :

statement(s)

Here statement(s) may be a single statement or a block of statements. The condition may be any expression, and true is any nonzero value. The loop iterates while the condition is true.

When the condition becomes false, program control passes to the line immediately following the loop. In Python, all the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses indentation as its method of grouping statements.

Example

```
# !/usr/bin/python
count=0
while (count < 9):
print 'The count is:',count
count=count +1
print "Good bye!"
```

OUTPUT

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
The count is: 5
The count is: 6
The count is: 7
```

```
The count is: 8
Good bye!
```