1 Shivani Raghuwanshi - [0108AI221059]

Saathi

Date __/__/__

Mid sem - II (2022-2023)

ITC 101 - Python Programming

Q.1 What are the various control statement in Python. Explain with example of each.

Ans: There are various types of control statement in Python:

(i) break statement
(ii) continue statement
(iii) pass statement

In python, loops are used to iterate repeatedly over a block of code. In order to change the way, a loop is executed from its usual behaviour, control statement.
Control statement are used to control the flow of the execution of the loop based on condition.

(i) break → the break statement is used to break out a ~~flow~~ for loop or a while loop.
In other words, the break statement terminate the current loop & resumes execution at the next statement.

Ex.
```
for i in range(8):
    if i > 3:
        break
    print(i)
```

Page No.

Shivani Raghuwanshi - [0108AI221059]

Saathi

Date __/__/__

(iv) continue → the continue statements rejects all the remaining statements in the ~~cur~~ ~~rent~~ current iteration of the loop & moves the control back to the ~~loop~~ top of the loop.

Ex.

```
for var in "python":
        if var = = "y" :
                continue
        print(var)
```

(vii) pass → when the pass statement is executed, nothing happens, but you avoid getting an error when empty code is not allowed. Empty code is not allowed in loops, function definitions, class definitions, or in if statements

Ex.

```
a = 10
b = 20
if (a < b):
    pass
else :
    print("b<a")
```

⇒ The pass statement is used as a placeholder for future code, it is a null operation ; nothing happens when it executes.

3 Shivani Raghuwanshi - [0108AI221059]

Saathi

Date __/__/__

Q.2 Observe the following python code very carefully & rewrite it after removing all syntactical errors with each correction underlined.

```
DEF execmain ( ):
        x = input ("Enter a number:")
        if (abs (x)  = x):
                print "You entered a positive number"
        else:
                x = * -1
                print "Number made positive:" x
execmain ( )
```

Ans:
```
def execmain ( ):
      x = input ("Enter a number")
      if (abs (x) == x):
            print ("You entered a positive number")
      else:
            x*= -1
            print ("Number made positive:", x)
execmain ( )
```

Q.3 Write the output of the following Python program code:

```
def ~~[illegible]~~ ChangedList( ):
    L =[ ]
    L1 =[ ]
    L2 =[ ]
~~[illegible]~~for i in range (1,10):
        L.append (i)
```

(4) Shivani Raghuwanshi - [0108AI221059]

Saathi

Date ___/___/___

```
for i in range (10, 1, -2):
    L1. append (i)
for i in range (len (L1)):
    L2 . append (L1[i] + L[i])
L2. append (len (L) - len (L1))
print (L2)
ChangeList ( )
```

Ans: [11, 10, 9, 8, 7, 4]

Q.4 Explain operator overloading with an example.

Ans: Operator overloading means giving extended meaning beyond their ~~[illegible]~~ predefined operational meaning.

For Ex. operator + is used to add two integers as well as join two strings & merge two lists. It is achievable because '+' operator is overloaded by int & str class. You might have noticed that the same built in operator or function shows different behaviour for objects of different classes, this is called operator overloading.

Ex.
```
class flower:
    def __init__ (self, rose, lotus):
        self. rose = rose
        self. lotus = lotus
    def __add__ (self, favorite):
        return self. rose + favorite. rose, self.
        lotus + favorite. lotus
```

(5) Shivani Raghuwanshi - [0108AI221059]

Saathi

Date __/__/____

```
object 1 = flower ("red" ,"pink")
object 2 = flower ("black", "white")
object3 = (object1 + object2)
print (object3)
```

Q.5 Write a user defined function to generate odd numbers between a & b (including b)

NOTE: a & b are received as an argument by the function

Ans:

```
def generate_odd_numbers (a,b):
        for num in range (a, b+1):
            if num%2 != 0 :
                print (num)
```

[copied from google]

Page No. ____