Name Ashank Agrawal
Sch. No. 327157
Year 2022-23
Date 7 June 2023

Question 01:- What are the various control statements in python. Explain with example of each.

Answer 01:- 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 statements are used to control the flow of the execution of the loop based on a condition. There are many types of control statements in python ~~and in this~~

Control statements in Pythons:-
Break statement
Continue statement
Pass statement

Break statement

The break statement in Python is used to terminate or abandon the loop containing the statement and brings the control out of the loop. It is used with both the while and the for loops, especially with nested loops to quit the loop. It terminates the inner loop and control shifts to the statement in the outer loop.

Inputs:-

```
age = input(" Please enter your age:")

while True:
    if int(age) >= 18:
        break
    else:
        print(" you are not eligible to vote")
```

outputs:-

Please enter your age : 17 you are not eligible to vote

Please enter your age : 18

In the above example, if the age entered is equal to or more than 18, it breaks out of the loop.

Continue statement:-

When a program encounters a Continue statement in Python, it skips the execution of the current iteration when the condition is met and lets the loop continue to move to the next iteration. It is used to continue running the program even after the program encounters a break during execution.

Inputs -

```
for letter in 'Flexiple':
if letter == '':
    Continue
print ('Letters:', letter)
```

outputs -

Letters : F
Letters : l
Letters : e
Letters : x
Letters : i
Letters : P
Letters : l
Letters : e

In this example, the program will skip the space " " in the world and continue with the rest of the iteration.

Pass statement:-

The pass statement is a null operator and is used when the programmer wants to do nothing when the condition is satisfied. This control statement in python does not terminate or skip the execution, it simply passes to the next iteration.

A loop cannot be left empty otherwise the interpreter will throw an error and to avoid this a programmer can use the pass statement.

Input:-

```
for letter in 'Flexiple':
    if letter == 'x':
        Pass
    print ('Letters:', Letter)
```

output:-

```
Letters: F
Letters: l
Letters: e
Letters: i
Letters: p
Letters: l
Letters: e
```

As you can see in the above example, even though the condition was met, the pass statement did not do anything and execution moved to the next iteration.

Name Ashank Agrawal
Sch. No. 32715
Year 2022-23
Date 7 June 2023

Question 02:- observe the following Python code very carefully and 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()
```

Answer:-

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

Question 03 - write the output of the following Python program Code:

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

Answer:-

output:- [11, 10, 9, 8, 7, 4]

Name Oshank Agrawal
Sch. No. 32715
Year 2022-23
Date 7 June 2022

Question 04:- Explain operator overloading with an example.

Answer 04-

Operator overloading:-

operator overloading means giving extended meaning beyond their predefined operational meaning. For example operator '+' is used to add two integers as well as join two strings and merge two lists. It is achievable because '+' operator is overloaded by int class and str class.

You might have noticed that the same built-in operator or function shows different behavior for objects of different classes, this is called operator Overloading.

~~example:- # + operator to show use of diff~~
~~print (1+2)~~

Example:-

```
# + operator for different purpose
print (1+2)  # to get sum of integer

print("OSHANK" + "AGRAWAL") # to Concatenate
# strings

print (3*4) # to get product of two numbers
```

print("OSHANK" * 4) # to Repeat the string.

Output:-

3
OSHANK AGRAWAL
12
OSHANKOSHANKOSHANKOSHANK

Question 05:- Write a user-defined function to generate odd numbers between a and b (including b)
Note:- a and b are received as an argument by the function.

Answer:-

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

def odd_num(a, b):
    for i in range(a, b+1):
        if i % 2 != 0:
            print(i)

odd_num(a, b)
```