

Chapter 9: Flow of Control

1. Which statement is used to take a decision in Python?
a) for b) if c) while d) def
2. How many times will the following loop run?
for i in range(2, 10, 2):
a) 4 b) 5 c) 8 d) 10
3. Which keyword skips the current iteration and moves to the next?
a) break b) pass c) continue d) exit
4. The 'else' block of a loop executes when:
a) loop breaks b) loop finishes normally c) never d) always first
5. range(5) generates values:
a) 1 to 5 b) 0 to 5 c) 0 to 4 d) 1 to 4
6. Which loop is best when the number of iterations is NOT known in advance?
a) for b) while c) range d) nested
7. What does the 'break' statement do?
a) skips one iteration b) ends the loop c) restarts loop d) nothing
8. An 'if' statement without an 'else' is called:
a) null if b) empty if c) one-way selection d) nested if
9. What will be the output of the following?
for i in range(1, 6): print(i, end=' ')
a) 1 2 3 4 5 b) 0 1 2 3 4 c) 1 2 3 4 d) Error
10. What will happen when the following code is executed?
x = 45
while x < 50: print(x)
a) Prints 45 to 49 b) Prints 45 once c) Infinite loop d) Error
11. Which of the following is an entry-controlled loop?
a) do-while b) for c) repeat-until d) None
12. What will range(2, 10, 3) generate?
a) 2, 3, 4...9 b) 2, 5, 8 c) 2, 4, 6, 8 d) 3, 6, 9
13. State True / False: The else clause of a loop executes when a break statement is used.
14. State True / False: The continue statement terminates the entire loop.
15. Fill in the blank: A loop becomes an infinite loop if the condition never becomes _____.
16. Find the output:
x = 10
y = 0
while x > y:
 print(x, y)
 x = x - 1
 y = y + 1
a) 10 0 b) Infinite c) 10 0, 9 1, 8 2, 7 3, 6 4 d) Error
17. Which of the following will create an infinite loop?
a) for i in range(10): pass
b) i = 0
 while i < 5: i += 1
c) while True: break
d) i = 1
 while i > 0: print(i)
18. What is the output of the following?
for i in range(5):
 if i == 3: break
 print(i)

- a) 0 1 2 b) 0 1 2 3 c) 1 2 3 d) Error
19. What is the output of the following?
for i in range(5):
 if i == 3: continue
 print(i)
a) 0 1 2 3 4 b) 0 1 2 4 c) 1 2 4 d) Error
20. What is the output of the following?
for i in range(1, 4):
 for j in range(1, 3):
 print(i, j)
a) 1 1 1 2 2 1 2 2 3 1 3 2 b) Error c) 1 1 2 2 3 3 d) None
21. What is the output of the following?
for i in range(1, 6):
 if i % 2 == 0: continue
 print(i)
a) 1 3 5 b) 2 4 c) 1 2 3 4 5 d) Error

SECTION B – Short Answer Questions

- What is the difference between a 'for' loop and a 'while' loop? When would you prefer each?
- Explain the working of break and continue statements with an example each.
- What is an infinite loop? Write a situation that causes it and how to avoid it.
- Differentiate between the if-elif-else and nested if statements.
- What is the purpose of the range() function? Explain its three parameters with examples.
range(stop) range(start, stop) range(start, stop, step)
- What is the difference between entry-controlled and exit-controlled loops? Give one example of each.
- Explain the three types of control flow in Python:
a) Sequence b) Selection c) Iteration
- Explain how the else clause of a loop behaves differently when a break statement is used.
- Rewrite the following for loop as an equivalent while loop:
for i in range(1, 11):
 print(i)
- Write the difference between the following:
a) break and continue
b) for loop and while loop
c) if-else and if-elif-else

SECTION C – Find the Output

- Predict the output:
for i in range(1, 6):
 print(i, end=' ')
- Predict the output:
for i in range(1, 10, 3):
 print(i)
- Predict the output:
i = 1
while i <= 5:
 if i == 3: break
 print(i)
 i += 1
- Predict the output:
for i in range(1, 6):
 if i % 2 == 0: continue
 print(i)

5. Predict the output (nested loop):

```
for i in range(1, 4):  
    for j in range(1, 3):  
        print(i, j)
```

6. Predict the output:

```
count = 0  
while count < 5:  
    print(count, end=' ')  
    count += 1
```

7. Predict the output:

```
for i in range(1, 6):  
    print(i)  
else:  
    print('Loop ended normally')
```

SECTION D – Find and Correct the Errors (for / while / if-else)

1. Find the errors and rewrite the correct code:

```
for i in range(5)  
    print(i)  
if x = 5:  
    print('five')
```

2. Find the errors and rewrite the correct code:

```
i = 1  
while i <= 5  
    print(i)  
    i += 1
```

3. Find the errors and rewrite the correct code:

```
n = 5  
for k in range(1, n)  
    if k = 3:  
        print(k)
```

4. Find the errors and rewrite the correct code:

```
for i in range(1, 11)  
    if i % 2 = 0:  
        print(i, 'Even')  
Else:  
    print(i, 'Odd')
```

5. Find the errors and rewrite the correct code:

```
i = 1  
while i <= 10  
    if i % 3 == 0  
        print(i, 'divisible by 3')  
    i += 1
```

6. Find the errors and rewrite the correct code:

```
for i in range(1, 6):  
    if i = 3:  
        break  
    Elif i == 4:  
        continue  
    print(i)
```

7. Find the errors and rewrite the correct code:

```
count = 0  
while count < 5  
    if count = 3:  
        print('Found 3')  
    else  
        print(count)  
    count += 1
```

8. Find the errors and rewrite the correct code:

```
a = int(input('Enter: '))  
if a > 0  
    print('Positive')  
Elif a < 0:  
    print('Negative')  
else:  
    print('Zero')
```

***** ALL THE BEST *****