PES UNIVERSITY

CELEBRATING 50 YEARS

Computing Fundamentals using Python

SUBJECT CODE : UQ25CA151A

Samyukta D Kumta

Computer Applications

Loop Control Statements

- Control statements modify the loop's execution flow.
- Python provides three primary control statements:
  - ✓ break,
  - ✓ continue and
  - ✓ pass.

Loop Control Statements

1. break Statement

The break statement is used to exit the loop prematurely

when a certain condition is met.

```
# Using break to exit the loop
for i in range(10):
    if i == 5:
        break
    print(i)
```

Output: 0 1 2 3 4

Computing Fundamentals using Python

Loop Control Statements

1. break Statement

The break statement is used to exit the loop prematurely

when a certain condition is met.

# Using break to exit the loop

```
for i in range(10):
    if i == 5:
        break
    print(i)
```

Output: 0 1 2 3 4

Explanation:

- The loop prints numbers from 0 to 9.
- When i equals 5, the break statement exits the loop.

Loop Control Statements

1. Continue Statement

The continue statement skips the current iteration and proceeds to the next iteration of the loop.

# Using continue

```
for i in range(10):
    if i % 2 == 0:
        continue
    print(i)
```

Output: 1   3   5   7   9

Computing Fundamentals using Python

Loop Control Statements

1. Continue Statement

The continue statement skips the current iteration and proceeds to the next iteration of the loop.

```
# Using continue
x = 0
while x < 5:
    x += 1
    if x == 2:
        continue  # skip when x is 2
    print(x)
```

Output: 1   3   4 5

Loop Control Statements

1. Pass Statement

The pass statement is used when a statement is syntactically required but we don’t want any code to execute.

```
# Using pass

for i in range(5):
    if i == 3:
        pass  # do nothing when i == 3
    else:
        print(i)
```

Output: 0   1   2   4

Loop Control Statements

1. Pass Statement

The pass statement is used when a statement is syntactically required but we don’t want any code to execute.

```
# Using pass

for i in range(5):
    if i == 3:
        pass  # do nothing when i == 3
    else:
        print(i)
```

Output: 0   1   2   4

Computing Fundamentals using Python

Loop Control Statements

Multiple choice questions

```
for i in range(5):
    if i == 3:
        pass
    print(i)
```

a) 0 1 2 3 4
b) 0 1 2 4
c) 1 2 3
d) Error

Ans: a

Computing Fundamentals using Python

Loop Control Statements

Multiple choice questions

```
for i in range(5):
    if i == 3:
        break
    print(i)
```

a) 0 1 2 3 4
b) 0 1 2
c) 3 4
d) Error

Ans: b

Multiple choice questions

Which of the following is TRUE?

a) pass and continue both skip the current iteration
b) break only works in while loop
c) pass is used to create placeholders in code
d) continue terminates the loop

Ans: c

Loop Control Statements

Write python code for the following scenarios

1. A teacher is taking attendance for roll numbers 1–10.
If a student’s roll number is 5, the teacher does nothing (just passes) but continues for others.

2. A company tracks working days (Mon–Sun).If the day is Saturday or Sunday, skip printing it.

3. A shopkeeper sells chocolates. Starting stock is 10.Each sale reduces stock by 1. If stock reaches 0, stop selling.

Loop Control Statements

Write python code for the following scenarios

4 An ATM allows **3 attempts** to enter the correct PIN (1234).
If the PIN is correct, print *Access Granted* and stop.
If 3 wrong attempts, stop program.

5. Create a menu:

1. Add
2. Subtract
3. Multiply
4. Exit

- If user enters an **invalid option** → pass (ignore).
- If user enters 2 → skip subtraction step (continue).
- If user enters 0 → exit menu (break).

THANK YOU

Samyukta D Kumta
Department of Computer Applications
samyuktad@pes.edu