

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Program	B.Pharmacy
Semester	First
Course Title /Subject Name	Basics of Python Programming for Pharmaceutical Sciences (Theory)
Course Code	BP101T
Module	Control Structures & Functions
Course Coordinator	Ms. Ramanjit Kaur
Mobile no	8968729307

Course Objectives:

The objectives of this course are to:

1. Introduce the fundamentals of python programming for pharmaceutical sciences.
2. Develop basic programming skills using control structures, functions ,and data structures.
3. Provide knowledge of data handling techniques for structured dataset management.
4. Familiarize students with data analysis tools such as NumPy and Pandas for healthcare datasets.
5. Enable students to visualize and interpret pharmaceutical data.

Course Outcomes (CO):

CO No.	Upon successful completion of this course, the students will be able to:
1	Explain the fundamentals of Python programming, including variables, data types, operators, and libraries.
2	Analyze program logic using control structures and functions.
3	Organize, manipulate, and retrieve data using data structures and file handling techniques.
4	Analyze pharmaceutical datasets using Python libraries.
5	Visualize and interpret pharmaceutical data using graphical tools.

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS
COLLEGE) BELA

LO	Learning Outcomes	Course Outcome Code
LO1	Understand Conditional Statements	BP101.2
LO2	Apply Looping Constructs	BP101.2
LO3	Use Break and Continue Statements	BP101.2
LO4	Develop and Use Functions	BP101.2
LO5	Write Modular Programs	BP101.2
LO6	Implement Pharmaceutical Applications	BP101.2
LO7	Problem-Solving and Computational Thinking	BP101.2

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Control Structures & Functions

Conditional statements

A conditional statement in Python is a programming statement used to make decisions based on conditions. It executes different blocks of code depending on whether a condition is True or False.

Example

```
age = 18
```

```
if age >= 18:  
    print("Eligible to vote")  
else:  
    print("Not eligible to vote")
```

In this example:

- If the condition `age >= 18` is true, the first message is printed.
- Otherwise, the `else` block runs.

Types of Conditional Statements in Python

1. `if`
2. `if-else`
3. `if-elif-else`
4. Nested if statements

The If Statement

An if statement is used as a form of conditional programming; something you probably do every day in the real world. That is, you need to decide whether you are going to have tea or coffee or to decide if you will have toast or a muffin for breakfast etc. In each of these cases you are making a choice, usually based on some information such as I had coffee yesterday, so I will have tea today.

In Python such choices are represented programmatically by the if condition statement. In this construct if some condition is true some action is performed, optionally if

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

it is not true some other action may be performed instead.

For Example

```
</> python
age = 20
if age >= 18:
    print("You are eligible to vote")
```

```
</> python
You are eligible to vote
```

The if-else statement

An **if-else statement** in Python is a conditional statement used to execute one block of code when a condition is **True** and another block of code when the condition is **False**.

For Example

```
a = 50
if a < 10: print("Hello World")
else:
    print("Python is Fun")
```

In this code, the expression `a < 10` will return False, so the else part will be executed. Therefore, the output will be “Python is Fun.”

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

The if-elif-else statement

The `if-elif-else` statement allows a program to choose one option from many conditions.

syntax

```
</> python
if condition1:
    # code block 1
elif condition2:
    # code block 2
else:
    # code block 3
```

Example

```
</> python
marks = 85
if marks >= 90:
    print("Grade A")
elif marks >= 70:
    print("Grade B")
else:
    print("Grade C")
The nested statement
```

Output

```
</> python
Grade B
```

Explanation

- If marks are 90 or above → Grade A
- Else if marks are 70 or above → Grade B
- Otherwise → Grade C

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Nesting If Statements

A nested statement in Python is a statement written inside another statement.

A nested if statement means placing one if statement inside another if statement to check multiple conditions.

Syntax

```
</> python
if condition1:
    # outer statement
    if condition2:
        # inner statement
```

Example

```
</> python
age = 20
citizen = True
if age >= 18:
    print("Person is adult")
    if citizen:
        print("Eligible to vote")
```

Output

```
</> python
Person is adult
Eligible to vote
```

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

LOOPS

A **loop** in Python is a programming statement used to repeat a block of code multiple times until a condition is satisfied.

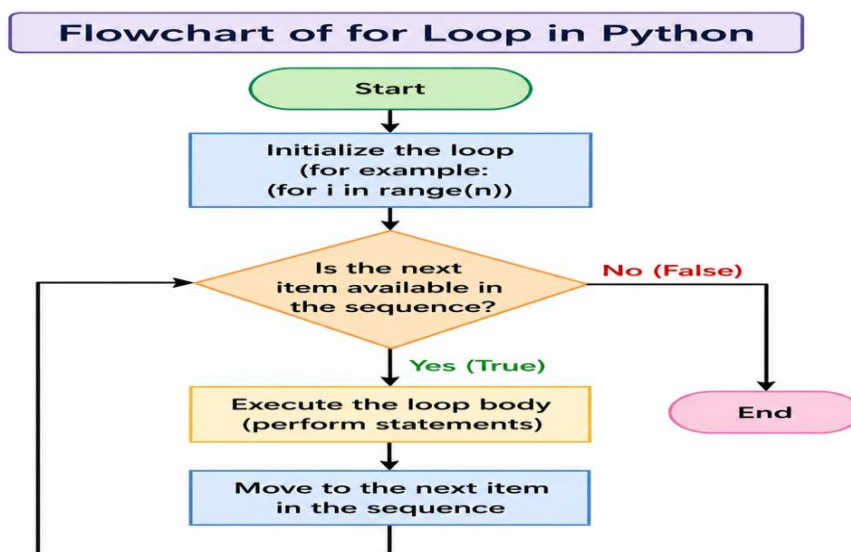
Types of Loops in Python

1. for loop
2. while loop

For Loop

A for loop is used to iterate over a sequence such as numbers, lists, strings, or tuples.

In many cases we know how many times we want to iterate over one or more statements (as indeed we did in the previous section). Although the while loop can be used for such situations, the for loop is a far more concise way to do this. It typically also clearer to another programmer that the loop must iterate for a specific number of iterations. The for loop is used to step a variable through a series of values until a given test is met. The behaviour of the for loop is illustrated in below.



ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

For Example

```
</> python
for i in range(1, 6):
    print(i)
```

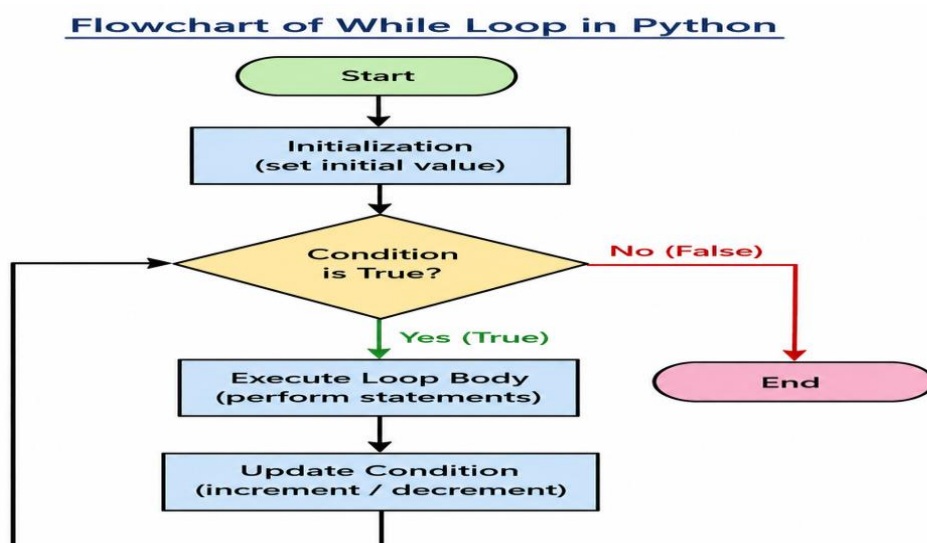
output

```
</> python
1
2
3
4
5
```

While Loop

The while loop exists in almost all programming languages and is used to iterative (or repeat) one or more code statements as long as the test condition (expression) is True. This iteration construct is usually used when the number of times we need to repeat the block of code to execute is not known. For example, it may need to repeat until some solution is found or the user enters a particular value.

The behaviour of the while loop is illustrated in below.



ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

EXAMPLE

```
</> python
num = 1
while num <= 3:
    print(num)
    num += 1
```

output

```
</> python
1
2
3
```

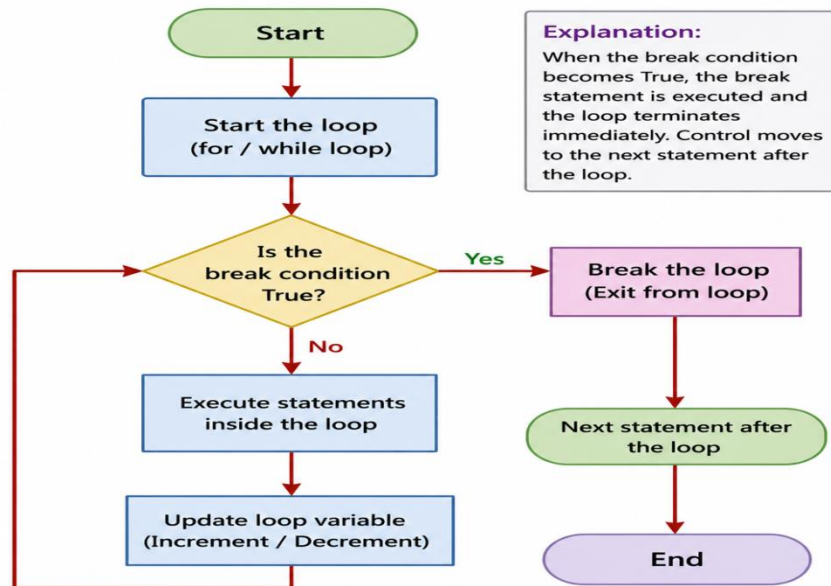
Break Statement

Python allows programmers to decide whether they want to break out of a loop early or not (whether we are using a for loop or a while loop). This is done using the break statement. The break statement allows a developer to alter the normal cycle of the loop based on some criteria which may not be predictable in advance (for example it may be based on some user input). The break statement, when executed, will terminate the current loop and jump the program to the first line after the loop.

The following diagram shows how this works for a for loop:

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Break Statement in Python (Flowchart)



This is shown below for a simple application where the user is asked to enter a number which may or may not be in the range defined by the for loop. If the number is in the range, then we will loop until this value is reached and then break the loop (that is terminate early without processing the rest of the values in the loop):

```
print('Only print code if all cycle completed')
num = int(input('Enter a number to check for: '))
for i in range(0, 6):
    if i == num:
        break
    print(i, ' ', end="")
print('Done')
```

If we run this and enter the value 7 (which is outside of the range) then all the values in the loop should be printed out:

```
Enter a number to check for: 7
0 1 2 3 4 5 Done
```

However, if we enter the value 3 then only the values 0, 1 and 2 will be printed out before the loop breaks (terminates) early:

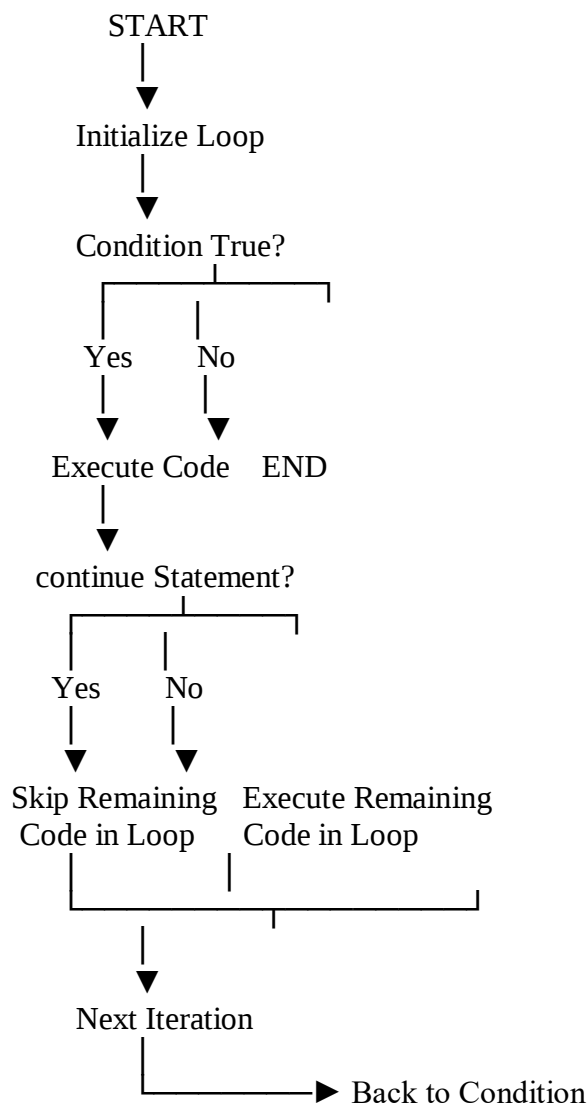
```
Enter a number to check for: 3
0 1 2 Done
```

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Note that the break statement can come anywhere within the block of code associated with the loop construct (whether that is a for loop or a while loop). This means that there can be statements before it and after it.

The continue statement

The continue statement also affects the flow of control within the looping constructs for and while. However, it does not terminate the whole loop; rather it only terminates the current cycle round the loop. This allows you to skip over part of a loop's cycle for a particular value, but then to continue with the remaining values in the sequence.



As with the break statement, the continue statement can come anywhere within the body of the looping construct. This means that you can have some statements that will be executed for every value in the sequence and some that are only executed when the continue statement is not run.

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

This is shown below. In this program the continue statement is executed only for odd numbers and thus the two print() statements are only run if the value of i is even:

```
for i in range(0, 10):  
    print(i, ' ', end='')  
    if i % 2 == 1:  
        continue  
    print('hey its an even number')  
    print('we love even numbers')  
    print('Done')
```

When we run this code we get

```
0 hey its an even number  
we love even numbers  
1 2 hey its an even number  
we love even numbers  
3 4 hey its an even number  
we love even numbers  
5 6 hey its an even number  
we love even numbers  
7 8 hey its an even number  
we love even numbers  
9 Done
```

As you can see, we only print out the messages about a number being even when the values are 0, 2, 4, 6 and 8.

Defining and Calling Functions in Python

A function is a block of reusable code that performs a specific task. Functions help make programs shorter, organized, and easier to understand.

1. Defining a Function

In Python, a function is defined using the `def` keyword.

Syntax

```
python
def function_name():
    code block
```

Example

```
</>python
def greet():
    print("Hello")
```

Here:

def → keyword used to create a function

greet → function name

print("Hello") → function body

2. Calling a Function

Calling a function means executing the function.

Example

```
</>python
def greet():
    print("Hello")
greet()
```

Output

```
text
Hello
```

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Passing Arguments to Functions

Arguments are values passed to a function when it is called.

Syntax

```
python
def function_name(parameter):
code
```

Example

```
</>python
def greet(name):
    print("Hello", name)

greet("Rahul")
```

Output
text

```
Hello Rahul
```

Here:

name is a parameter
"Rahul" is an argument

Returning Values from Functions

The return statement sends a value back from the function.

Example

```
</>python
def add(a, b):
    return a + b

result = add(5, 3)
print(result)
```

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Output

```
text
8
```

Here:

return a + b returns the sum
result stores the returned value

Complete Example

```
</>python
def multiply(x, y):

    return x * y

answer = multiply(4, 5)
print("Multiplication =", answer)
```

Output

text

```
Multiplication = 20
```

Here is a simple example of a **modular Python program** for pharmaceutical applications involving:

1. **Dosage Calculation**
2. **BMI Calculation**

The program uses **functions (modules)** to make the code reusable, organized, and easy to maintain.

Modular Program for Pharmaceutical Applications

</> Python

```
# Pharmaceutical Application Program
```

```
# Function to calculate dosage
def calculate_dosage(weight, dose_per_kg):
    dosage = weight * dose_per_kg
    return dosage
```

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

```
# Function to calculate BMI
def calculate_bmi(weight, height):
    bmi = weight / (height ** 2)
    return bmi

# Function to determine BMI category
def bmi_category(bmi):
    if bmi < 18.5:
        return "Underweight"
    elif bmi < 24.9:
        return "Normal weight"
    elif bmi < 29.9:
        return "Overweight"
    else:
        return "Obese"

Main program
def main():
    print("PHARMACEUTICAL APPLICATION")
    print("1. Dosage Calculation")
    print("2. BMI Calculation")

    choice = int(input("Enter your choice (1 or 2): "))

    if choice == 1:
        weight = float(input("Enter patient weight (kg): "))
        dose_per_kg = float(input("Enter dose per kg (mg/kg): "))

        dosage = calculate_dosage(weight, dose_per_kg)

        print("Required Dosage =", dosage, "mg")

    elif choice == 2:
        weight = float(input("Enter weight (kg): "))
        height = float(input("Enter height (m): "))

        bmi = calculate_bmi(weight, height)

        print("BMI =", round(bmi, 2))
    print ("Category =", bmi_category(bmi))
```

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

```
else:  
    print("Invalid choice")
```

```
# Calling main function  
main()
```

Explanation of Modules

1. Dosage Calculation Module

Uses the formula:

$\text{Dosage} = \text{Weight} \times \text{Dose per kg}$

Example:

- Weight = 60 kg
- Dose = 5 mg/kg
- Dosage = 300 mg

2. BMI Calculation Module

Uses the formula:

$\text{BMI} = \frac{\text{Weight (kg)}}{\text{Height (m)}^2}$

Example:

- Weight = 70 kg
- Height = 1.75 m
- BMI = 22.86

Features of the Program

- Modular design using functions
- Reusable code
- Easy to maintain and debug
- User-friendly menu system
- BMI classification included

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Output

PHARMACEUTICAL APPLICATION

1. Dosage Calculation
2. BMI Calculation

Enter your choice (1 or 2): 1

Enter patient weight (kg): 60

Enter dose per kg (mg/kg): 5

Required Dosage = 300 mg

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Multiple Choice Questions (MCQs) – Control Structures & Functions

1. Which statement is used for decision making in Python?

- A) loop
- B) if
- C) break
- D) continue

Answer: B) if

2. Which keyword is used for an alternative condition?

- A) else
- B) elif
- C) while
- D) def

Answer: A) else

3. Which statement checks multiple conditions?

- A) if
- B) while
- C) if-elif-else
- D) break

Answer: C) if-elif-else

4. What is a nested condition?

- A) Loop inside loop
- B) Function inside function
- C) if statement inside another if statement
- D) None

Answer: C) if statement inside another if statement

5. Which loop is used when the number of iterations is known?

- A) while loop
- B) for loop
- C) nested loop
- D) infinite loop

Answer: B) for loop

6. Which loop works until a condition becomes false?

- A) for loop
- B) while loop
- C) nested loop
- D) break loop

Answer: B) while loop

7. Which keyword exits a loop immediately?

- A) stop
- B) exit
- C) break
- D) continue

Answer: C) break

8. Which keyword skips the current iteration?

- A) pass
- B) continue
- C) break
- D) skip

Answer: B) continue

9. What is the output of:

```
for i in range(3):  
    print(i)
```

- A) 1 2 3
- B) 0 1 2
- C) 0 1 2 3
- D) Error

Answer: B) 0 1 2

10. Which function is used to define a function in Python?

- A) function
- B) define
- C) def
- D) fun

Answer: C) def

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

11. What is the correct way to call a function?

- A) function_name[]
- B) function_name()
- C) call function_name
- D) run function_name

Answer: B) function_name()

12. What are parameters in functions?

- A) Loop variables
- B) Values passed to a function
- C) Conditions
- D) Outputs

Answer: B) Values passed to a function

13. Which keyword returns a value from a function?

- A) break
- B) continue
- C) return
- D) pass

Answer: C) return

14. What is the output?

x =5

if x >2:

print("Yes")

- A) No
- B) Yes
- C) Error
- D) Nothing

Answer: B) Yes

15. Which symbol is used for equality comparison?

- A) =
- B) ==
- C) !=
- D) =

Answer: B) ==

16. What is the output?

x =10

if x <5:

print("A")

else:

print("B")

- A) A
- B) B
- C) Error
- D) Nothing

Answer: B) B

17. Which loop can create an infinite loop?

- A) for
- B) while
- C) if
- D) elif

Answer: B) while

18. What is the output?

for i in range(1,4):

print(i)

- A) 0 1 2
- B) 1 2 3
- C) 1 2 3 4
- D) Error

Answer: B) 1 2 3

19. Which statement is optional in an if condition?

- A) if
- B) else
- C) condition
- D) indentation

Answer: B) else

20. What is indentation in Python?

- A) Decoration
- B) BSpaces before statements
- C) Variable
- D) Loop

Answer: B) Spaces before statements

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

21. Which function calculates BMI?

- A) bmi()
- B) calculate_bmi()
- C) body()
- D) weight()

Answer: B) calculate_bmi()

22. BMI stands for:

- A) Body Mass Index
- B) Body Measure Indicator
- C) Basic Mass Index
- D) None

Answer: A) Body Mass Index

23. BMI formula is:

$$\text{BMI} = \frac{\text{Weight}}{\text{Height}^2}$$

- A) $\text{Weight} \times \text{Height}$
- B) $\text{Weight} \div \text{Height}^2$
- C) $\text{Height} \div \text{Weight}$
- D) $\text{Weight} + \text{Height}$

Answer: B) $\text{Weight} \div \text{Height}^2$

24. Dosage calculation depends on:

- A) Weight
- B) Dose per kg
- C) Both A and B
- D) None

Answer: C) Both A and B

25. Dosage formula is:

$$\text{Dosage} = \text{Weight} \times \text{Dose per kg}$$

- A) $\text{Weight} + \text{Dose}$
- B) $\text{Weight} \times \text{Dose per kg}$
- C) $\text{Dose} \div \text{Weight}$
- D) None

Answer: B) $\text{Weight} \times \text{Dose per kg}$

26. Which loop is best for iterating through a list?

- A) while
- B) for
- C) if
- D) else

Answer: B) for

27. What is the output?

```
x = 7
if x % 2 == 0:
    print("Even")
else:
    print("Odd")
```

- A) Even
- B) Odd
- C) Error
- D) Nothing

Answer: B) Odd

28. Which keyword defines a block in Python?

- A) {}
- B) begin-end
- C) indentation
- D) brackets

Answer: C) indentation

29. Which statement is used inside loops?

- A) break
- B) continue
- C) pass
- D) All of these

Answer: D) All of these

30. What is the output?

```
for i in range(2):
    print("Python")
```

- A) Python
- B) Python Python
- C) Infinite output
- D) Error

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Answer: B) Python Python

31. What is a function?

- A) Variable
- B) Reusable block of code
- C) Loop
- D) Operator

Answer: B) Reusable block of code

32. Which loop checks the condition first?

- A) while
- B) for
- C) both
- D) none

Answer: A) while

33. Which value is returned if no return statement is used?

- A) 0
- B) None
- C) False
- D) Error

Answer: B) None

34. What is the output?

```
x = 3
if x > 5:
    print("A")
elif x > 2:
    print("B")
```

- A) A
- B) B
- C) Error
- D) Nothing

Answer: B) B

35. Which function can take arguments?

- A) Only built-in functions
- B) User-defined functions
- C) Both
- D) None

Answer: C) Both

36. Which keyword creates a loop?

- A) for
- B) while
- C) both
- D) none

Answer: C) both

37. What is the output?

```
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

Answer: A) 0 1 2

38. What is the output?

```
for i in range(5):
    if i == 2:
        continue
    print(i)
```

- A) 0 1 2 3 4
- B) 0 1 3 4
- C) 1 2 3 4
- D) Error

Answer: B) 0 1 3 4

39. Which condition is checked first in if-elif-else?

- A) else
- B) elif
- C) if
- D) none

Answer: C) if

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

40. Which loop is suitable when iterations are unknown?

- A) for
- B) while
- C) nested
- D) none

Answer: B) while

41. Which operator means “not equal to”?

- A) ==
- B) !=
- C) <=
- D) =

Answer: B) !=

42. What is the output?

```
def add(a,b):  
    return a+b
```

```
print(add(2,3))
```

- A) 23
- B) 5
- C) Error
- D) None

Answer: B) 5

43. What is the output?

```
x =5
```

```
while x >0:  
    print(x)  
    x = x -1
```

- A) 5 4 3 2 1
- B) 1 2 3 4 5
- C) Infinite loop
- D) Error

Answer: A) 5 4 3 2 1

44. Which statement does nothing?

- A) break
- B) continue
- C) pass
- D) return

Answer: C) pass

45. Which keyword is used to stop a function?

- A) stop
- B) break
- C) return
- D) continue

Answer: C) return

46. Which data type is returned by comparison operators?

- A) int
- B) float
- C) bool
- D) string

Answer: C) bool

47. What is modular programming?

- A) Writing long code
- B) Dividing program into functions/modules
- C) Using loops only
- D) None

Answer: B) Dividing program into functions/modules

48. Which of the following is a conditional statement?

- A) if
- B) while
- C) for
- D) def

Answer: A) if

49. Which function is useful in pharmaceutical dosage calculation?

- A) calculate_dosage()
- B) weight()
- C) medicine()
- D) health()

Answer: A) calculate_dosage()

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

50. What is the main advantage of functions?

- A) Increase code repetition
- B) Reduce readability

C) Reusability of code

D) Increase errors

Answer: C) Reusability of code

Short Questions and Answers

1. What is a conditional statement in Python?

A conditional statement is used to make decisions based on conditions.

2. What is the syntax of an if statement?

if condition:

statement

3. What is the purpose of the else statement?

The else statement executes code when the if condition is false.

4. What is the use of elif in Python?

The elif statement checks multiple conditions one by one.

5. What is a nested condition?

A nested condition is an if statement inside another if statement.

6. What is a loop?

A loop is used to repeat a block of code multiple times.

7. What is a for loop?

A for loop is used when the number of iterations is known.

8. What is a while loop?

A while loop repeats code while a condition is true.

9. What is the purpose of the break statement?

The break statement stops the loop immediately.

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

10. What is the purpose of the continue statement?

The continue statement skips the current iteration and moves to the next iteration.

11. What is a function in Python?

A function is a reusable block of code that performs a specific task.

12. Which keyword is used to define a function?

The def keyword is used to define a function.

13. How do you call a function?

A function is called using its name followed by parentheses. Example:

```
function_name()
```

14. What are function arguments?

Arguments are values passed to a function.

15. What is the use of the return statement?

The return statement sends a value back from a function.

16. What is modular programming?

Modular programming means dividing a program into small functions or modules.

17. What is BMI?

BMI (Body Mass Index) measures body weight according to height.

18. Write the BMI formula.

$$\text{BMI} = \frac{\text{Weight (kg)}}{\text{Height (m)}^2}$$

19. What is dosage calculation?

Dosage calculation determines the amount of medicine required for a patient.

20. Write the dosage calculation formula.

$$\text{Dosage} = \text{Weight} \times \text{Dose per kg}$$

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Long Questions:

1. Explain different types of conditional statements in Python with syntax and examples.
2. Describe nested conditional statements in Python. Write a program to demonstrate nested if statements.
3. Explain the for loop in Python with syntax, flowchart, and suitable examples.
4. Explain the while loop in Python. Differentiate between for loop and while loop with examples.
5. What are break and continue statements in Python? Explain their working with suitable programs.
6. Define functions in Python. Explain how to define and call functions with examples.
7. Explain passing arguments and returning values in Python functions with suitable examples.
8. Write a modular Python program for pharmaceutical dosage calculation and explain its working.
9. Write a modular Python program to calculate BMI and determine BMI categories.
10. Compare break and continue statements with suitable examples.

.

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Control Structures & Functions

Case Based Learning (CBL) and Problem Based Learning (PBL)

Case Based Learning (CBL)

Case Study 1: Medicine Eligibility Check Using Conditional Statements

Case:

A doctor wants to check whether a patient is eligible for a medicine. The medicine can only be given if the patient's age is above 18 years.

Python Program

```
python id="mj1wm3" age = int(input("Enter patient age:"))  
if age >= 18: print("Patient is eligible for medicine") else: print("Patient is not eligible for  
medicine")
```

Q1. What is the purpose of this program?

Answer: To check whether a patient is eligible for a medicine based on age.

Q2. Which statement is used for decision making?

Answer: if-else statement.

Q3. What condition is checked in the program?

Answer: age >= 18

Q4. Which variable stores the patient's age?

Answer: age

Q5. What data type is used for age?

Answer: Integer (int).

Q6. What message is displayed if the age is 20?

Answer: "Patient is eligible for medicine".

Q7. What message is displayed if the age is 15?

Answer: "Patient is not eligible for medicine".

Q8. What is the learning outcome of this case study?

Answer: Understanding if-else statements and decision making in healthcare applications.

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Case Study 2: Temperature Monitoring Using Nested Conditions

Case:

A hospital monitoring system checks patient temperature and classifies it as normal, fever, or high fever.

Python Program

```
python id="fmn89m"
temp = float(input("Enter body temperature: "))

if temp > 37:
    if temp > 39:
        print("High Fever")
    else:
        print("Fever")
else:
    print("Normal Temperature")
```

Q1. What is the purpose of this program?

Answer: To classify a patient's temperature as normal, fever, or high fever.

Q2. Which variable stores the body temperature?

Answer: temp

Q3. What data type is used for temperature?

Answer: Float (float).

Q4. What is a nested condition?

Answer: An if statement inside another if statement.

Q5. When is "High Fever" displayed?

Answer: When the temperature is greater than 39°C.

Q6. When is "Fever" displayed?

Answer: When the temperature is greater than 37°C but not greater than 39°C.

Q7. When is "Normal Temperature" displayed?

Answer: When the temperature is 37°C or below.

Q8. What is the learning outcome of this case study?

Answer: Understanding nested conditions and real-life healthcare monitoring.

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Problem Based Learning (PBL)

Problem Scenario 1: Dosage Calculation Program

Problem Statement

A pharmacist needs a program that calculates the correct medicine dosage based on a patient's weight.

Rule:

- If weight is less than 20 kg $\rightarrow$ Dosage = 5 mg/kg
- If weight is 20 kg or more $\rightarrow$ Dosage = 10 mg/kg

Learning Objectives

- Use conditional statements (if-else)
- Define and call functions
- Pass arguments and return values

Python Program

```
</python>
def calculate_dosage(weight):
    if weight < 20:
        dosage = weight * 5
    else:
        dosage = weight * 10
    return dosage

weight = float(input("Enter patient weight (kg): "))
dose = calculate_dosage(weight)

print("Required dosage:", dose, "mg")
```

Sample Output

Enter patient weight (kg): 15
Required dosage: 75 mg

ASBASJSM COLLEGE OF PHARMACY (AN AUTONOMOUS COLLEGE) BELA

Q1. What is the purpose of this program?

Answer: **To calculate the correct medicine dosage based on a patient's weight.**

Q2. What function is defined in the program?

Answer: **calculate_dosage()**

Q3. What argument is passed to the function?

Answer: **weight**

Q4. What condition is checked in the program?

Answer: **weight < 20**

Q5. What dosage is given if the weight is less than 20 kg?

Answer: **5 mg/kg.**

Q6. What dosage is given if the weight is 20 kg or more?

Answer: **10 mg/kg.**

Q7. What does the **return** statement do?

Answer: **It returns the calculated dosage value.**

Q8. Which statement is used for decision making?

Answer: **if-else statement.**

Q9. What data type is used for weight?

Answer: **Float (float).**

Q10. What is the output for a weight of 15 kg?

Answer: **Required dosage: 75 mg.**