

# Python Programming

Class : 11<sup>th</sup>

Subject : Computer Science and  
Entrepreneurship

## Chapter : 2

1. What should you do when installing Python to run it from the command line more easily?

Options:

- a) Uncheck "Add Python to PATH"
- b) Choose a different IDE
- ✓ c) Check "Add Python to PATH"
- d) Install only the IDE

**Explanation:**

Jab aap Python install karte hain, to "Add Python to PATH" ka option check karna chahiye, taake Python **command prompt (CMD)** se kahin se bhi run ho sake.

---

2. Which of the following is a valid variable name in Python?

Options:

- ✓ (a) variable1
- (b) 1variable
- (c) variable-name
- (d) variable name

**Explanation:**

Valid variable names:

- **Alphabets ya underscore se start** karte hain

- Spaces aur special characters (like -) allow nahi hote  
variable1 sahi hai; 1variable, variable-name, variable name sab invalid hain.
- 

### 3. What is the output of the below code?

```
age = 25  
print(" Age : ", age)
```

#### Options:

- ✓ (a) Age: 25
- (b) 25
- (c) Age
- (d) age

#### Explanation:

`print(" Age : ", age)` ka output: Age : 25 hota hai. Spaces and value dono properly print honge.

---

### 4. Which of the following operators is used for exponentiation in Python?

#### Options:

- (a) \*
- ✓ (b) \*\*
- (c) //
- (d) /

#### Explanation:

Python mein \*\* ka matlab **power/exponentiation** hota hai.

e.g.,  $2 ** 3 = 8$

---

### 5. Which loop type in Python is used to iterate over a collection such as lists?

#### Options:

- (a) while
- ✓ (b) for
- (c) do-while
- (d) repeat

**Explanation:**

Python mein **for loop** collections jaise lists, tuples, strings par iterate karne ke liye use hota hai.

---

**6. What does the `range()` function do in Python?****Options:**

- (a) Generates a list of numbers
- ✓ (b) **Creates a sequence of numbers**
- (c) Calculates the sum of numbers
- (d) Prints a range of numbers

**Explanation:**

`range()` ek **sequence of numbers** generate karta hai, jise loop mein use kiya jata hai.  
e.g., `range(5) → 0, 1, 2, 3, 4`

---

**7. Which keyword is used to define a function in Python?****Options:**

- (a) define
- (b) function
- ✓ (c) **def**
- (d) func

**Explanation:**

Python mein function banane ke liye **def keyword** ka use hota hai.

e.g., `def my_function():`

**8. What is the output of the below code?**

```
temperature, humidity, wind_speed = 25, 60, 15
print("Hot and humid" if temperature > 30 and humidity > 50 else
      "Warm and breezy" if temperature == 25 and wind_speed > 10 else
      "Cool and dry" if temperature < 20 and humidity < 30 else
      "Moderate")
```

**Options:**

- (a) Hot
- ✓ (b) **Warm**
- (c) Cool
- (d) Nothing

**Explanation:**

Code conditional expression check karta hai:

- `temperature == 25` ✓
  - `wind_speed > 10` ✓
- So, "Warm and breezy" print hoga.
- 

**09. Which operation is used to combine two lists?****Options:**

- (a) combine()
- (b) concat()
- ✓ (c) +
- (d) merge()

**Explanation:**

Python mein 2 lists ko combine karne ke liye + **operator** use hota hai.

```
list1 = [1, 2]
list2 = [3, 4]
print(list1 + list2) # Output: [1, 2, 3, 4]
```

---

**10. Which of the following steps is NOT part of the basic programming process?****Options:**

- (a) Write Code
- (b) Compile/Interpret
- (c) Execute
- ✓ (d) Ignore Errors

**Explanation:**

Error ko **ignore** karna kabhi bhi programming ka hissa nahi hota.  
Har programmer ko errors ko **fix/debug** karna hota hai.

## Short Question

**1. Explain the purpose of using comments in Python code?****Answer:**

Comments are used to make the code more readable by adding explanations. They help programmers understand the logic of the code and are ignored during program execution.

In Python, comments start with #.

**Example:**

```
# This is a comment
print("Hello")  # Prints Hello
```

---

2. Describe the difference between integer and float data types in Python. Provide an example of each.

**Answer:**

- **Integer (int):** Whole numbers without a decimal point.  
**Example:** `x = 10`
  - **Float:** Numbers with a decimal point.  
**Example:** `y = 10.5`
- 

3. Define operator precedence and give an example of an expression where operator precedence affects the result.

**Answer:**

Operator precedence defines the order in which operations are performed in an expression.

**Example:**

```
result = 5 + 2 * 3
```

Output is 11 because \* has higher precedence than +.

---

4. How does the shorthand if-else statement differ from the regular if-else statement?

**Answer:**

The shorthand if-else is written in a single line, making the code concise.

**Example:**

- **Regular:**

```
if a > b:
    print("A")
else:
    print("B")
```

- **Shorthand:**

```
print("A") if a > b else print("B")
```

---

### 5. Explain the use of the `range()` function in a for loop?

**Answer:**

The `range()` function generates a sequence of numbers and is commonly used in `for` loops to repeat a block of code.

**Example:**

```
for i in range(3):  
    print(i)  
# Output: 0 1 2
```

---

### 6. Explain how default parameters work in Python functions.

**Answer:**

Default parameters are used in function definitions to provide default values. If no value is passed, the default is used.

**Example:**

```
def greet(name="Guest"):  
    print("Hello", name)  
greet() # Output: Hello Guest
```

---

### 7. Explain why modular programming is useful in Python.

**Answer:**

Modular programming divides the program into small, reusable functions or modules. It makes code easier to read, test, maintain, and debug.

---

### 8. Explain the difference between a class and an object in Python.

**Answer:**

- **Class:** A blueprint for creating objects. It defines attributes and behaviors.
- **Object:** An instance of a class, representing a specific example.

**Example:**

```
class Car:  
    def start(self):  
        print("Car started")  
  
my_car = Car() # Object  
my_car.start()
```

## Long Question

### 1. Evaluate the following Python expressions

(a)

Expression: Evaluating Python Expressions

$$(18 / 3 + 4 ** 2) - (2 * (7 - 3)) / (9 - 7 + 4)$$

Step-by-Step Evaluation

1. Evaluate the expressions inside the parentheses:

$$18 / 3 = 6.0$$

$$4 ** 2 = 16$$

$$7 - 3 = 4$$

$$9 - 7 + 4 = 6$$

2. Evaluate the expressions with the calculated values:

$$(18 / 3 + 4 ** 2) = 6.0 + 16 = 22.0$$

$$(2 * (7 - 3)) = 2 * 4 = 8$$

3. Finally, evaluate the entire expression:

$$(18 / 3 + 4 ** 2) - (2 * (7 - 3)) / (9 - 7 + 4) = 22.0 - 8 / 6$$

$$8 / 6 = 1.3333 \text{ (approximately)}$$

$$22.0 - 1.3333 = 20.6667 \text{ (approximately)}$$

Final Result

The final result of the expression is approximately:

20.67

---

(b)

Expression:

$$(25 + 3 * 4 ** 2 - 6) / (2 ** 3 + 1) - 7$$

Step-by-step:

$$= (25 + 3 \times 16 - 6) / (8 + 1) - 7$$

$$= (25 + 48 - 6) / 9 - 7$$

$$= 67 / 9 - 7$$

$$\approx 7.44 - 7$$

✓ Answer: 0.44

---

(c)

Expression:

$$(12 + 6 * (5 - 2)) ** 2 / ((4 ** 2 - 7) + 10)$$

$$= (12 + 6 \times 3)^2 / (16 - 7 + 10)$$

$$= (12 + 18)^2 / 19$$

$$= 30^2 / 19$$

$$= 900 / 19$$

✓ Answer:  $\approx 47.37$

---

(d)

Expression:

$$45 / (2 ** 2 + 3 * 4) + 8 * (7 - 3)$$

$$= 45 / (4 + 12) + 8 \times 4$$

$$= 45 / 16 + 32$$

$$\approx 2.81 + 32$$

✓ Answer:  $\approx 34.81$

---

(e)

Expression:

$$(14 * 3 + 5 ** 2 - 28) / (4 + 2 ** 3) + 10 * 7.3$$

Step-by-Step:

Numerator:

- $14 * 3 = 42$
- $5 ** 2 = 25$
- $42 + 25 - 28 = 39$

### Denominator:

- $2 ** 3 = 8$
- $4 + 8 = 12$

### First part:

- $39 / 12 = 3.25$

### Second part:

- $10 * 7.3 = 73$

### Final Result:

- $3.25 + 73 = \checkmark 76.25$

---

## 2. Translate the following Mathematical Expressions to Python Syntax

(a)

$$5 \times (3 + 2^2) / 6 - 2 \times 3$$

### ✓ Python Syntax:

```
result = (5 * (3 + 2 ** 2)) / (6 - 2 * 3)
```

### □ Explanation:

- $*$  = multiplication
- $**$  = exponentiation (power)
- $/$  = division
- Parentheses  $()$  ensure correct **order of operations**

### Optional: Evaluation

Let me solve it too (if needed):

1.  $2 ** 2 = 4$
2.  $3 + 4 = 7$
3.  $5 * 7 = 35$
4.  $2 * 3 = 6$
5.  $6 - 6 = 0$

✗ Division by 0 is **not allowed** in Python → This will cause an error.

### ! Final Note:

This expression will cause a **ZeroDivisionError** in Python.

---

(b)

**Mathematical Expression:**

$$7 + 2^2$$

✓ **Python Syntax:**

```
result = 7 + 2 ** 2
```

🔍 **Evaluation:**

1.  $2 ** 2 = 4$
2.  $7 + 4 = \checkmark 11$

✓ **Final Answer: 11**

---

### 3. Explain the concept of variables in Python.

#### **Answer:**

Variables are names used to store data in memory so that it can be reused or modified later in a program.

They are containers for storing values.

#### **Rules:**

- Must begin with a letter or underscore.
- Cannot start with a digit.
- Cannot use reserved keywords.

#### **Example:**

```
age = 18  
name = "Ali"
```

Variables can hold different types: `int`, `float`, `str`, etc.

---

4. Write a Python program that takes a number as input and checks whether it is positive, negative, or zero using an if-elif-else statement.

Ans:

```
# Take input from user

num = float(input("Enter a number: "))

# Check the value

if num > 0:

    print("The number is positive.")

elif num < 0:

    print("The number is negative.")

else:

    print("The number is zero.")
```

Outputs:

**Input:** 12

**Output:** The number is positive.

**Input:** -5.6

**Output:** The number is negative.

**Input:** 0

**Output:** The number is zero.

6. Write a Python program using a while loop that prints all the odd numbers between 1 and Also, count and print the total number of odd numbers.

Ans:

```
# Initialize variables

num = 1

count = 0
```

```
print("Odd numbers between 1 and 100 are:")

while num <= 100:

    if num % 2 != 0:

        print(num, end=" ")

        count += 1

    num += 1

print("\nTotal number of odd numbers between 1 and 100:", count)
```

**Output (shortened):**

```
Odd numbers between 1 and 100 are:
1 3 5 7 9 ... 97 99
Total number of odd numbers between 1 and 100: 50
```