

Student Learning Outcomes

By the end of this chapter, you will be able to:

- Understand basic programming concepts and set up a Python development environment.
- Write and interpret basic Python syntax and structure, including variables, data types, and input/output operations.
- Use various operators and expressions in Python, including arithmetic, comparison assignment, logical operators and operator precedence.

Subjective Type Questions

Long Questions prepared in the light of new Examination Techniques
(Knowledge, Understanding, Application, Analytical & Conceptual)

Introduction

Q. Explain briefly the concept of python?

Ans: Welcome to the world of Python programming! Python is a popular language known for its simplicity and readability, making it ideal for both beginners and professionals. In this chapter, we will start with the basics, including setting up your development environment and learning fundamental programming concepts. By the end of this chapter, you will have a comprehensive understanding of Python, enabling you to write, test, and debug your own programs.

3.1**Introduction to Python Programming**

Q.1. What is python language. Describe various fields of python applications. Also describe advantage of python over different languages

Ans: Define:

Python is a widely-used high-level programming language famous for its simplicity and readability.

Applications of Python: It is versatile language and applicable to various fields, including web development, data analysis and artificial intelligence etc.

Advantage of Python: Whether you are creating a simple script or developing a complex software, Python's user-friendly nature helps you get started quickly and efficiently.

Q.2. Define Computer Programming. Also give an example.

Ans:

Computer programming is the process of creating a set of instructions that tell a

computer how to perform a task. These instructions are written in a programming language that the computer can understand and execute.

Example: Installing mobile app from play store when we click on install button of any mobile app, we give clear and precise instructions to the computer to perform specific tasks.

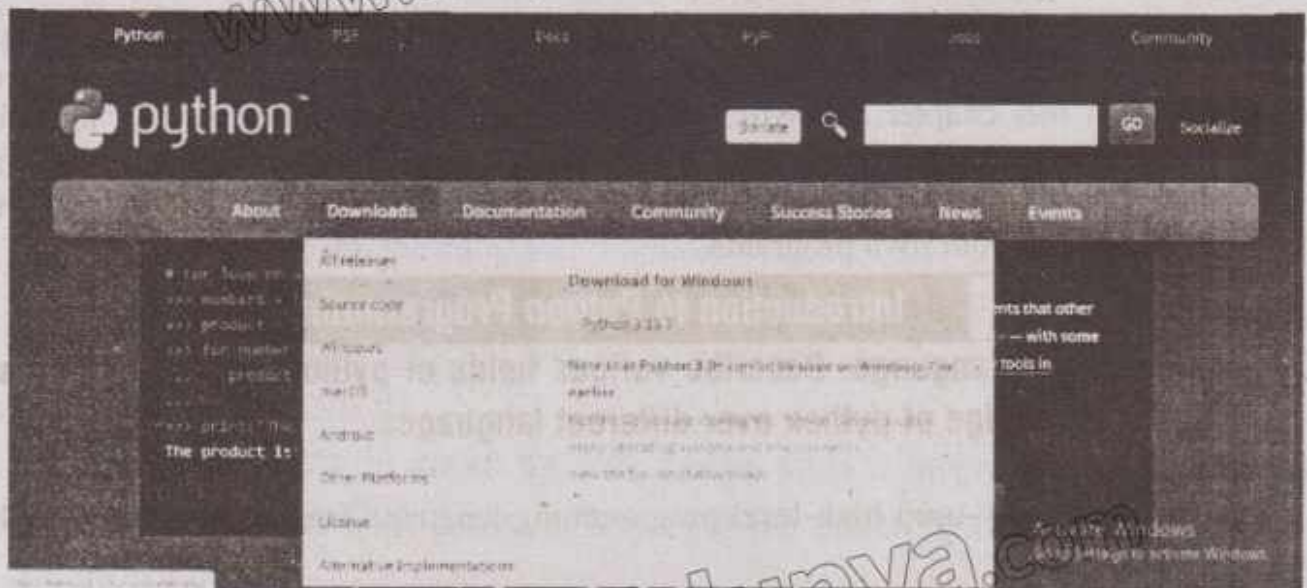
TOOL TIP

When installing Python, make sure to check the box that says "Add Python to PATH." This makes it easier to run Python from the command line. We can also use online services to write and run Python program.

Q.3. What is setting up python development environment? Also describe the procedure to set python IDE?

Ans: The development environment refers to the process of preparing a computer to write, run, and debug Python code effectively. This involves installing and configuring the necessary software, tools, and libraries that make development smoother and more efficient. We can download and install Python from <https://www.python.org/>

1. Download Python: Visit <https://www.python.org/>. Navigate to the "Downloads" section and click on the latest version of Python (e.g., Python 3.13.x) for your operating system (Windows, macOS, or Linux).



The installer will download automatically.

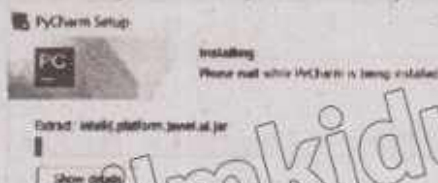
2. Install Python:

- Run the downloaded .exe file.
- Check the box "Add Python to PATH" during installation.
- Select "Install Now" or customize the installation if needed (e.g., choose the installation directory).
- Wait for the installation to complete, then click "Close".



3. Set Up an IDE:

- Download and install a popular IDE like PyCharm, VS Code, or IDLE (included with Python).
- PyCharm is a powerful IDE for Python development, available in a free Community edition or a paid Professional edition. Visit <https://www.jetbrains.com/pycharm/> to download the version suitable for your operating system (Windows, macOS, or Linux).



4. Configuring the IDE with Python:

- Open PyCharm and start a new project.
- In the "Configure" or "Settings" menu, go to "Project Interpreter."
- Click the gear icon, select "Add Interpreter," and choose the path to your Python executable (e.g., C:\Python312\python.exe on Windows or /usr/bin/python3 on Linux/macOS). Browse to your installation folder if it's not auto-detected.
- Apply the settings and let PyCharm index the interpreter.



4. Testing the configuring:

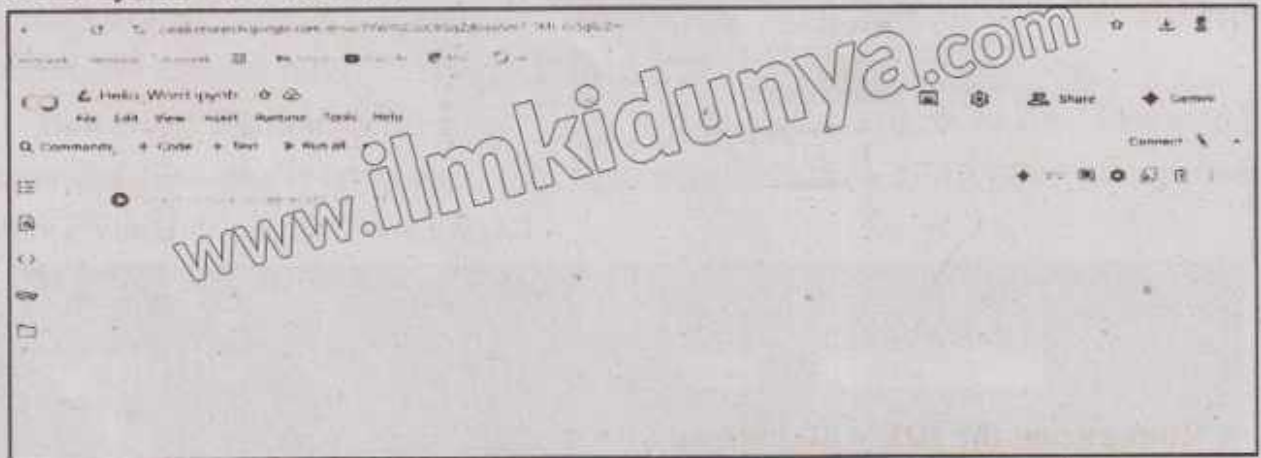
- Create a new file (e.g., test.py) in your IDE.
- Write a simple script: `print("Hello, Python!")`.
- Run the script (use the "Run" button in PyCharm/IDLE or F5 in VS Code after setting up a task).
- If the output appears (e.g., "Hello, Python!"), your IDE is correctly configured.

Q.4. What is google colab? Also write the method to use google colab

Ans: Another excellent option for using Python is Google Colab. It is a free cloud-based platform that provides a Jupyter notebook environment with pre-installed Python and popular libraries like NumPy, Pandas, and Matplotlib.

How to use google colab:

To get started, visit <https://colab.research.google.com/>, sign in with a Google account, and create a new notebook. No local installation is required, making it ideal for beginners or those working on machines with limited resources. You can write and execute Python code in cells.



3.2

Basic Python Syntax and Structure

Q.5. Describe basic python syntax and structure?

Ans:

Python language is popular due to its simplicity and readability. Look at the following example:

```
print("Hello! World")
```

Explanation: In this example, the `print` function is utilized to output the message enclosed in double quotation marks. This illustrates Python's straightforward syntax, where functions like `print` are used to perform actions, such as displaying text.

Q.6. Describe python comments? Also give example of each its type.

Ans:

Comments are the lines that are not executed by the Python interpreter. They are used to provide explanations or notes for the Programmer. In python, two types of comments are used:

- Single line comments
- Multi-line comments

Single line comments:

Single-line comments start with the # symbol. This type of comment is used only to add explanation remarks only in one line.

Example:

```
print ("K2 is the second-highest mountain in the world")
#This line shows one message.
```

Multi-line comments: Multi-line comments start with the ''' symbol and also end with ''' symbol. This type of comment is used when explanation is written in more than one line.

Example:

```
'''
This is a comment.
'''

print ("Edhi Foundation operates the world's largest
volunteer ambulance network.")
```

INTERESTING INFORMATION

We can use "..." (Triple Quotes) in Python for multiline comments but actually they are multiline string literals which are ignored at runtime.

Q.7. Define variable in python? Also write the rules to name a variable.

Ans: Definition:

A variable in programming functions as a storage container within a computer's memory. It allows the storage of data that can be retrieved later in the code.

Why it is named as variable:

The value of a variable can change throughout the execution of a program, which is why it is referred to as a "variable":

Example:

```
age = "Hello, dear students!"
print(age)
age = 71
print ("Azam lived for", age, "years")
age = 60
print ("Iqbal lived for", age, "years")
```

Output:

```
Azam lived for 71 years
Iqbal lived for 60 years
```

Explanation:

This example illustrates how the variable "age" is assigned different values to represent the ages of Azam and Iqbal, which are then printed as part of the output.

Variable Naming Rules in Python:

Variable names in Python must adhere to the following rules:

Starting:

The name must begin with a letter (a-z, A-Z) or an underscore (_).

Subsequent characters:

Subsequent characters can include letters, digits (0-9), or underscores (_).

Variable names:

Variable names are case-sensitive, meaning age and Age are considered two different variables.

Reserved Words:

Python's reserved keywords, such as for, while, if, etc., cannot be used as variable names.

DO YOU KNOW

Following are some of the reserved words in Python:

- if, else, elif, for, while, break, continue, and, or, not, in, is, True, False, None.
- We cannot use them as a variable name.

Q.8. Describe the types of variables used in python.

Ans: Types of variables:

In Python, you can create variables of different types to store various kinds of data. Here are some common types of variables:

- **Integer (int):**

This data type is used to store whole numbers.

Example: age = 17

- **Floating-point (float):**

This data type is used to store decimal numbers.

Example: Price = 19.99

- **String (str):**

It is the default data type used in python. This data type is used to store text.

Example: name = "Ali"

- **Boolean (bool):**

This data type is used to store boolean values i.e. True or False.

Example: is_student = True

Q.9. Describe the input and output functions used in python.

Ans:

Input and output operations allow you to interact with the user. You can ask the user to enter data (input) and display information to the user (output).

Input:

We use the input () function to get user input. The input () function displays a message on the screen and waits for the user to type something and press Enter. The text entered by the user is then stored in a variable.

Example:

Code:

```
name = input("Enter your name: ")  
print("Hello, " + name + "!")
```

Output:

```
Enter your name: Maryam Ashraf  
Hello, Maryam Ashraf!
```

Explanation:

This line of code asks the user to enter their name and stores it in the variable name.

Output:

Use the print() function to display information on the screen. The print() function takes one or more arguments and displays them

Example:

Code:

```
name = input("Enter your name: ")  
print("Hello, " + name + "!")
```

Output:

```
Enter your name: Maryam Ashraf  
Hello, Maryam Ashraf!
```

Explanation:

This line of code displays a greeting message that includes the user's name.

Q.10. Describe the integer and float input in python.

Ans: Handling Integer and Float Inputs:

To handle numeric inputs, you use the int() or float() functions to convert input strings to integers or floating-point numbers.

Integer Inputs:

```
user_age = int(input("Enter your age: "))  
print("Your age is:", user_age)
```

Output:

```
Enter your age: 16  
Your age is: 16
```

Float Inputs:

```
user_height = float(input("Enter your height in meters: "))  
print("Your height is", user_height, "meters")
```

Output:

```
Enter your height in meters: 1.5  
Your height is 1.5 meters
```

DO YOU KNOW
In Python indentation defines structure and scope of your code. Incorrect indentation may cause errors in your code.

3.3

Operators and Expressions

Q.11. Define operator and expression. Also explain types of operators in detail.

Ans: Operators:

Operators are the symbols which are used to perform different operations on operands.

Expression: An expression is a combination of operators, operands and values to produce the results.

Types of Operators:

Arithmetic Operators: Arithmetic operators are used to perform basic mathematical operations such as addition, subtraction, multiplication, division, modulus, exponentiation, and floor division.

Example:

Define variables

a = 10

b = 3

Perform all arithmetic operations on these numeric variables and print results

print(a, "+", b, "=", a + b) # Output: 10 + 3 = 13

print(a, "*", b, "=", a * b) # Output: 10 * 3 = 30

print(a, "/", b, "=", a / b) # Output: 10 / 3 = 3.3333333333333335

print(a, "//", b, "=", a // b) # Floor division # Output: 10 // 3 = 3

print(a, "%", b, "=", a % b) # Modulus # Output: 10 % 3 = 1

print(a, "**", b, "=", a ** b) # ** represents power operator # Output: 10 ** 3 = 1000

Comparison Operators:

Comparison operators are used to compare two values or expressions. They determine the relational logic between them, such as equality, inequality, greater than, less than, and so on. These operators return a Boolean value (True or False) based on the comparison result.

Example:

Define variables

x = 10

y = 5

Greater than

print(x, ">", y, "=", x > y) # Output: 10 > 5 = True

Less than

print(x, "<", y, "=", x < y) # Output: 10 < 5 = False

Equal to

```
print(x, "==", y, "=", x == y)
```

```
# Output: 10 == 5 = False
```

Not equal to

```
print(x, "!=", y, "=", x != y)
```

```
# Output: 10 != 5 = True
```

Greater than or equal to

```
print(x, ">=", y, "=", x >= y)
```

```
# Output: 10 >= 5 = True
```

Less than or equal to

```
print(x, "<=", y, "=", x <= y)
```

```
# Output: 10 <= 5 = False
```

Explanation:

The above code demonstrates Python's comparison operators by defining two variables, x and y, and comparing them using various operators like >, <, ==, !=, >=, and <=.

Assignment Operator:

Assignment operators are used to assign values to variables. The most common assignment operator is the equal sign (=), which assigns the value on the right to the variable on the left.

Compound Assignment Operator:

There are also compound assignment operators like +=, -=, *=, /=, %=, and **=, which combine arithmetic operations with assignment.

Example:

Define initial values

```
a = 10
```

```
b = 5
```

Assignment

```
x = a; print("a = ", x)
```

```
# Output: a = 10
```

Addition assignment

```
a += b; print("a after addition = ", a)
```

```
# Output: a = 15
```

Subtraction assignment

```
a -= b; print("a after subtraction = ", a)
```

```
# Output: a = 10
```

Multiplication assignment

```
a *= b; print("a after multiplication = ", a)
```

```
# Output: a = 50
```

Division assignment

```
a /= b; print("a after division = ", a)
```

```
# Output: a = 10.0
```

Floor division assignment

```
a //= b; print("a after floor division = ", a)
```

```
# Output: a = 2.0
```

Modulus assignment

```
a %= b; print("a after modulus = ", a)
```

```
# Output: a = 2.0
```

Exponentiation assignment

```
a **= b; print("a after exponentiation = ", a)
```

```
# Output: a = 32.0
```

Explanation:

The above code illustrates the use of compound assignment operators in Python. It

starts by assigning a value to a variable "a". It then demonstrates various assignment operations including addition, subtraction, multiplication, division, floor division, modulus, and exponentiation. Each operation updates the value of "a" and the results are printed with the updated values. The code comments show the output of each operation.

Logical Operators:

Logical operators are used to combine multiple conditions or expressions. The most common logical operators are "and", "or", and "not"; they are used to perform logical operations and return Boolean values based on the evaluation of the expressions involved.

Example:

Define variables

```
x = True
```

```
y = False
```

Logical AND

```
logical_and = x and y
```

```
print(x, "and", y, "=", logical_and)
```

Output: True and False = False

Logical OR

```
logical_or = x or y
```

```
print(x, "or", y, "=", logical_or)
```

Output: True or False = True

Logical NOT

```
logical_not_x = not x
```

```
print("not", x, "=", logical_not_x)
```

Output: not True = False

```
logical_not_y = not y
```

```
print("not", y, "=", logical_not_y)
```

Output: not False = True

Explanation:

The above code demonstrates the use of logical operators in Python. It defines two Boolean variables, x and y, and performs logical operations such as "and", "or", and "not" on them. The results of these logical operations are printed, showing whether the expressions evaluate to True or False. The code also includes comments with the output of each logical operation.

Q.12. Define operator precedence in python. Describe the sequence of operator precedence in python. Also give example.

Ans: Operator Precedence:

Operator precedence determines the order in which operations are performed in an expression. In Python as well as in Mathematics, certain operators have higher precedence and are evaluated before others. Understanding this helps ensure that your calculations are done correctly.

Sequence of Operators in Python:

1. **Parentheses ():** Highest precedence. Operations inside parentheses are performed first.

$(3 + 2) * 4$ evaluates to 20.

2. **Exponentiation:** ****** Performs power operations next.

$2 + 2 * 2 ** 3 = 18$ (not 32)

3. **Multiplication *, Division / and Modulus %:** These operations come next, evaluated from left to right.

4. **Addition + and Subtraction - :** These have lower precedence compared to multiplication and division.

Example:

$4 * 3 + 10 / 2 - 11 \% 3$

- $4 * 3$ evaluates to 12
- $10 / 2$ evaluates to 5.0
- $11 \% 3$ evaluates to 2
- Then perform addition and subtraction: $12 + 5.0 - 2 = 15.0$

Example: Consider the following code to show operator precedence:

```
print("=== PYTHON OPERATOR PRECEDENCE DEMONSTRATION ===\n")
```

1. Arithmetic Operators

```
print("1. ARITHMETIC OPERATORS:")
```

```
print(f'3 + 2 * 5 = {3 + 2 * 5}') # Multiplication before addition
```

```
print(f'(3 + 2) * 5 = {(3 + 2) * 5}') # Parentheses change order
```

```
print(f'2 ** 3 * 4 = {2 ** 3 * 4}') # Exponentiation before multiplication
```

```
print(f'15 / 3 * 2 = {15 / 3 * 2}') # Division and multiplication (left to right)
```

```
print(f'15 // 4 + 2 = {15 // 4 + 2}') # Floor division before addition
```

```
print(f'11 // 4 + 2 = {11 // 4 + 2}') # Floor division before addition
```

```
print(f'17 % 5 * 2 = {17 % 5 * 2}') # Modulo before multiplication
```

Output:

```
=== PYTHON OPERATOR PRECEDENCE DEMONSTRATION ===
```

1. ARITHMETIC OPERATORS:

$3 + 2 * 5 = 13$

$(3 + 2) * 5 = 25$

$2 ** 3 * 4 = 32$

$15 / 3 * 2 = 10.0$

$15 // 4 + 2 = 5$

$11 // 4 + 2 = 4$

$17 \% 5 * 2 = 4$

DO YOU KNOW

In Python "f" before string is used for string interpolation (embedding variables and expressions) inside a string.

TIDBIT

Exponentiation is right associative, meaning that when there are multiple exponentiation operators, Python evaluates them from right to left.

Objective Type Questions

Additional MCQ's

Multiple Choice Questions prepared in the light of new Examination Techniques
(Knowledge, Understanding, Application, Analytical & Conceptual)

Introduction

1. Python is popular due to:
(A) Simplicity (B) Readability (C) Easy understanding (D) All above
2. Finding problems in Python code is called:
(A) Test (B) Debug (C) Checking (D) None of them

3.1

Introduction to Python Programming

3. As compare to other languages Python is:
(A) More complex (B) Easy to Debug
(C) Quickly to Execution (D) Efficient Language
4. Why python more successful now a days?
(A) It is easy to understand (B) Difficult to handle complex problems
(C) Versatile applications (D) Professional Language
5. Which of the following Python Programming have?
(A) Complex structure (B) Ability to handle complex structure
(C) Beginners language (D) Professional language
6. Process to create set of instructions:
(A) Software (B) Programming (C) Execution (D) Debugging
7. Prepare Computer to run and debug Python code is called:
(A) Running Python (B) Setting Python
(C) Develop Environment (D) Debugging Python
8. Install software tool means:
(A) Configuring necessary tools
(B) Configuring necessary tools & libraries
(C) Configuring necessary libraries (D) Configuring OS libraries
9. What is the Advantage of development environment?
(A) Make running & creating code efficient
(B) Help developer to easy debug
(C) Help to handle large projects (D) Above all
10. Which makes it easier to run Python from command line?
(A) Source code (B) Add Python to Path
(C) Object code (D) Add Python Library
11. Who is latest version of Python is?
(A) Python 3.11x (B) Python 3.12x (C) Python 3.13x (D) Python 3.14x

12. Python latest version is compatible for:
(A) Window (B) MacOS (C) Linux (D) All
13. For installation user need to run _____ file?
(A) .dat (B) .exe (C) .doc (D) .setup
14. Installation directory refers to:
(A) Python Library (B) Python Directory
(C) Saving Python source code location (D) None of them
15. Popular Python IDE is:
(A) PyCharm (B) VS Code (C) IDLE (D) All above
16. Which of the following is paid professional edition for Python?
(A) PyCharm (B) VS Code (C) IDLE (D) Above all
17. Which icon is used in PyCharm to add interpreter?
(A) New Project icon (B) Gear icon
(C) New Script icon (D) None of them
18. Python use _____ translator:
(A) Assembler (B) Compiler (C) Interpreter (D) All
19. Which short key is used to run script in both PyCharm and IDLE?
(A) F1 (B) F5 (C) F2 (D) F9
20. Google Colab is a:
(A) Search engine (B) Cloud base platform
(C) Web browser (D) Python IDE
21. Google Colab provide _____ environment:
(A) MatPlotLib (B) IDE
(C) Jupyter Notebook (D) None of them
22. Matplotlib is a:
(A) Compiler (B) IDE (C) Interpreter (D) Built-in Library

3.2

Basic Python Syntax and Structure

23. In Python, print() is a:
(A) Function (B) Keyword (C) Variable (D) Library
24. In print statement written inside double quotes is called:
(A) Message (B) String (C) Paragraph (D) Syntax
25. Which of the following is a Symbol of single line comments?
(A) # (B) "" (C) ! (D) //
26. Which type of comments need to closed?
(A) Single line (B) Multiline (C) Both (D) None
27. Multi-line comment is also called:
(A) Multiline string (B) Multiline string literals
(C) Multiline command (D) None of them
28. Which of the following is not a variable?
(A) variable11 (B) 11variable (C) Var (D) Air

29. A variable can have alphabets, digits and _____.
 (A) symbol (B) characters (C) reports (D) underscore
30. In Python programming age and Age are:
 (A) Two different variable (B) Two same variables
 (C) Consume same memory (D) Have different data type
31. In input function, if data-type of variable is not mentioned, then Python considered it as _____ type?
 (A) int (B) float (C) string (D) bool
32. Which of the following is an output function?
 (A) print() (B) input() (C) int() (D) float()
33. To handle numeric input we use:
 (A) int() (B) float() (C) Both a, b (D) None of them

3.3

Operator & Expressions

34. Guess the output of $10 \% 3$?
 (A) 0 (B) 3 (C) 1 (D) 10
35. Symbol of integral division:
 (A) / (B) // (C) % (D) **
36. Which of the following is a power operator?
 (A) / (B) // (C) % (D) **
37. Comparison operator gives answer in:
 (A) Integer form (B) Exponential form
 (C) Floating point (D) Boolean form
38. In comparison operator $!=$ is called:
 (A) Not equal (B) Equal (C) Small equal (D) Large equal
39. Which of the following is an assignment operator?
 (A) $!=$ (B) $==$ (C) $=$ (D) $>=$
40. If $a=10$, $b=3$. What is the output of $a = b$?
 (A) 100 (B) 1000 (C) 10000 (D) 100000
41. Logical operators are used to:
 (A) Solve expression (B) Analyze results
 (C) Combine multiple expression (D) None of them
42. In Python logical-and operator represented by:
 (A) and (B) or (C) not (D) $\&\&$
43. In Python logical-or represented by:
 (A) and (B) $\&\&$ (C) or (D) $\parallel$
44. Which of the following have highest precedence?
 (A) ** (B) * (C) / (D) %
45. Which of the following is right associate?
 (A) * (B) () (C) + (D) **

Answers

- | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|
| 1. D | 2. A | 3. D | 4. C | 5. B | 6. B | 7. C |
| 8. B | 9. D | 10. B | 11. C | 12. D | 13. B | 14. C |
| 15. D | 16. A | 17. B | 18. C | 19. B | 20. B | 21. C |
| 22. D | 23. A | 24. B | 25. A | 26. B | 27. B | 28. B |
| 29. D | 30. A | 31. C | 32. A | 33. C | 34. C | 35. B |
| 36. D | 37. D | 38. A | 39. C | 40. B | 41. C | 42. A |
| 43. C | 44. A | 45. D | | | | |

Additional Short Question's

Short Answer Questions prepared in the light of new Examination Techniques
(Knowledge, Understanding, Application, Analytical & Conceptual)

Introduction

1. Define Python.

Ans: Python is a popular language known for its simplicity and readability, making it ideal for both beginners and professionals.

3.1

Introduction to Python Programming

2. Write down the applications of python?

Ans: It is versatile language and applicable to various fields, including:

- Web development
- Data analysis
- Artificial intelligence etc.

3. What are the advantage of python?

Ans: Whether you are creating a simple script or developing a complex software, Python's user-friendly nature helps you get started quickly and efficiently.

4. Define computer programming.

Ans: Computer programming is the process of creating a set of instructions that tell a computer how to perform a task.

Example: Installing mobile app from play store

5. Which option must be check when installing python and why?

Ans: When installing Python, make sure to check the box that says "Add Python to PATH." This makes it easier to run Python from the command line. We can also use online services to write and run Python program.

6. Write down the procedure to install python.

Ans: Following is a procedure to install python

- Run the downloaded .exe file.
- Check the box "Add Python to PATH" during installation.

- Select "Install Now" or customize the installation if needed (e.g., choose the installation directory).
- Wait for the installation to complete, then click "Close".

7. Define google colab.

Ans: It is a free cloud-based platform that provides a Jupyter notebook environment with pre-installed Python and popular libraries like NumPy, Pandas, and Matplotlib.

8. How google colab is used online?

Ans: To get started, visit <https://colab.research.google.com/>, sign in with a Google account, and create a new notebook. No local installation is required, making it ideal for beginners or those working on machines with limited resources. You can write and execute Python code in cells.

3.2

Basic Python Syntax and Structure

9. Explain the basic python syntax and structure.

Ans: Python language is popular due to its simplicity and readability. Look at the following example:

```
print("Hello! World")
```

Explanation: In this example, the print function is utilized to output the message enclosed in double quotation marks. This illustrates Python's straightforward syntax, where functions like print are used to perform actions, such as displaying text.

10. Define single line comment with example.

Ans: Single-line comments start with the # symbol. This type of comment is used only to add explanation remarks only in one line.

Example:

```
print("K2 is the second-highest mountain in the world")
#This line shows one message.
```

11. Define Multiline comment with example.

Ans: Multi-line comments start with the ''' symbol and also end with ''' symbol. This type of comment is used when explanation is written in more than one line.

Example:

```
'''
This is a comment.
'''
```

```
print("Edhi Foundation operates the world's largest
volunteer ambulance network.")
```

12. Why it is compulsory to close multi-line comment?

Ans: It is compulsory to close multi-line comment because if we don not close it python translator include all bottom lines as a comment and these lines does not get executed which means that output will not display as you desired.

13. Why variable name used in python? Give an example.

Ans: A variable in programming functions as a storage container within a computer's memory. It allows the storage of data that can be retrieved later in the code.

Example:

```
age = "Hello, dear students!"
print(age)
age = 71
print ("Azam lived for", age, "years")
```

14. Write down any two rules to name a variable.

Ans: Variable names in Python must adhere to the following rules:

- The name must begin with a letter (a-z, A-Z) or an underscore (_).
- Variable names are case-sensitive, meaning age and Age are considered two different variables.

15. Write any two types of variables.

Ans:

- **Integer (int):**

This data type is used to store whole numbers.

Example: age = 17

- **Floating-point (float):**

This data type is used to stores decimal numbers.

Example: Price = 19.99

16. Explain the purpose of input and output functions.

Ans: Input and output operations allow you to interact with the user. You can ask the user to enter data (input) and display information to the user (output).

17. Differentiate between integer and float datatype.

Ans:

Integer Data Type	Float Data Type
Represents whole numbers (positive, negative, or zero)	Represents real numbers that require a decimal point or are expressed in exponential form.
integers have unlimited precision	Floats have double precision (typically 64-bit)
Using the floor division operator // returns an integer.	Any arithmetic operation involving a float will typically produce a float result

3.3

Operator & Expressions

18. Define operator and expression.

Ans: **Operators:** Operators are the symbols which are used to perform different operations on operands.

Expression: An expression is a combination of operators, operands and values to produce the results.

19. Define Arithmetic operators.

Ans: Arithmetic operators are used to perform basic mathematical operations such as addition, subtraction, multiplication, division, modulus, exponentiation, and floor division.

20. Write the example of floor division, modulus and power operator.

Ans:

a = 10

b = 3

print(a, "//", b, "=", a // b) # Floor division # Output: 10 // 3 = 3

print(a, "%", b, "=", a % b) # Modulus # Output: 10 % 3 = 1

print(a, "**", b, "=", a ** b) # ** represents power operator # Output: 10 ** 3 = 1000

21. Define Comparison operator.

Ans: Comparison operators are used to compare two values or expressions. They determine the relational logic between them, such as equality, inequality, greater than, less than, and so on. These operators return a Boolean value (True or False) based on the comparison result.

22. Why logical operators are used?

Ans: Logical operators are used to combine multiple conditions or expressions. The most common logical operators are "and", "or", and "not"; they are used to perform logical operations and return Boolean values based on the evaluation of the expressions involved.

23. What is compound assignment operator give example.

Ans: There are also compound assignment operators like +=, -=, *=, /=, %=, and **=, which combine arithmetic operations with assignment.

Example:

Define initial values

a = 10

b = 5

Addition assignment

a += b; print("a after addition =", a) # Output: a = 15

Subtraction assignment

a -= b; print("a after subtraction =", a) # Output: a = 10

24. Define operator precedence.

Ans: Operator precedence determines the order in which operations are performed in an expression. In Python as well as in Mathematics, certain operators have higher precedence and are evaluated before others. Understanding this helps ensure that your calculations are done correctly.

25. Solve $8+3*4**2$

Ans: Step-by-step:

1. Exponentiation first: $2**2=4$

Expression becomes: $8+3*4-4$

2. **Multiplication next:** $3*4=12$

Expression becomes: $8+12-4$

3. **Addition and Subtraction** (from left to right).

$$8+12=20$$

$$20-4=16$$

Result: 16

Exercise Questions (Solved)



Multiple Choice Questions

- Which of the following steps is NOT part of the basic programming process?
(a) Write Code (b) Compile/Interpret (c) Execute (d) Ignore Errors
- What should you do when installing Python to run it from the command line more easily?
(a) Uncheck "Add Python to PATH" (b) Choose a different IDE
(c) Check "Add Python to PATH" (d) Install only the IDE
- What will be the output of: `print(10 // 3)`?
(a) 3.333 (b) 3 (c) 1 (d) 4
- What is the data type of variable `x` in `x = "123"`?
(a) Integer (b) Float (c) String (d) Boolean
- Which operator has the highest precedence in Python?
(a) Addition (+) (b) Multiplication (*)
(c) Exponentiation (**) (d) Parentheses ()
- What is the result of `5 > 3` and `2 < 1`?
(a) True (b) False (c) Error (d) None
- What does the `+=` operator do?
(a) Adds two numbers (b) Compares values
(c) Adds and assigns (d) Multiplies values
- What is the output of the below code?
`age = 25; print(" Age : ", age)`
(a) Age: 25 (b) 25 (c) Age (d) age
- Which symbol is used for comments in Python?
(a) // (b) /* (c) # (d) --

Answers

- | | | | | |
|--------|--------|--------|--------|--------|
| 1. (d) | 2. (c) | 3. (b) | 4. (c) | 5. (d) |
| 6. (b) | 7. (c) | 8. (a) | 9. (c) | |

Short Questions

1. Why are comments important in programming?

Ans: Comments are important because:

- They help us to understand the logic of our code to other users.
- They help us to understand our own code after the years of writing it.

2. Discuss three basic data types in python.

Ans:

- **Integer (int):**

This data type is used to store whole numbers.

Example: age = 17

- **Floating-point (float):**

This data type is used to store decimal numbers.

Example: Price = 19.99

- **String (str):**

It is the default data type used in python. This data type is used to store text.

Example: name = "Ali"

3. Describe the difference between integer and float data types in python.

Ans:

Integer Data Type	Float Data Type
Represents whole numbers (positive, negative, or zero)	Represents real numbers that require a decimal point or are expressed in exponential form.
Integers have unlimited precision	Floats have double precision (typically 64-bit)
Using the floor division operator // returns an integer.	Any arithmetic operation involving a float will typically produce a float result

4. What are logical operators and name all three?

Ans: Logical operators are used to combine multiple conditions or expressions. The most common logical operators are "and", "or", and "not"; they are used to perform logical operations and return Boolean values.

Types of Logical operators:

- And operator (and)
- Or operator (or)
- Not operator (not)

5. How do you get user input in python?

Ans: We use the input () function to get user input. The input () function displays a message on the screen and waits for the user to type something and press Enter. The text entered by the user is then stored in a variable.

Example:

Code:

```
name = input("Enter your name: ")
print("Hello, " + name + "!")
```

Output:

Enter your name: Maryam Ashraf
Hello, Maryam Ashraf!

6. What is the purpose of print() statement in python?

Ans: Use the print() function to display information on the screen. The print() function takes one or more arguments and displays them

Example:**Code:**

```
name = input("Enter your name: ")  
print("Hello, " + name + "!")
```

Output:

Enter your name: Maryam Ashraf
Hello, Maryam Ashraf!

7. What is operator precedence? Give an example.

Ans: Operator precedence determines the order in which operations are performed in an expression. In Python as well as in Mathematics, certain operators have higher precedence and are evaluated before others

$4 * 3 + 10 / 2 - 11 \% 3$

- $4 * 3$ evaluates to 12
- $10 / 2$ evaluates to 5.0
- $11 \% 3$ evaluates to 2
- Then perform addition and subtraction: $12 + 5.0 - 2 = 15.0$

8. Write any three main rules for naming variables in python?

Ans:

Starting:

The name must begin with a letter (a-z, A-Z) or an underscore (_).

Subsequent characters:

Subsequent characters can include letters, digits (0-9), or underscores (_).

Variable names:

Variable names are case-sensitive, meaning age and Age are considered two different variables.

Long Questions**1. What are the basic data types in python? Explain integers, floats, strings, and Booleans with examples.**

Ans: For Answer See Q No. 8.

2. What are arithmetic operators in python. Provide examples of each operator and explain when you would use them.

Ans: For Answer See Q No. 11.

3. Explain the input() and print() functions with examples.

Ans: For Answer See Q No. 9.

4. Explain how comparison and logical operators work together in conditional statements. Provide a practical example

Ans: For Answer See Q No. 4

5. Write a program in python that will take a name from user and display greeting message to the given name.

Ans: Code:

```
# Program to take user's name and display a greeting message
```

```
# Taking input from the user
```

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

```
# Displaying the greeting message
```

```
print("Hello, " + name + "! Welcome to Python programming.")
```

6. Write a program in python that takes two numbers from user and display their sum.

Ans: Code:

```
num1 = input("Enter first number: ")
```

```
num2 = input("Enter second number: ")
```

```
print("The sum is: " + num1 + num2)
```

7. Write a program in python that five numbers from user and display their sum.

Ans: Code:

```
# Program to take five numbers from user and display their average
```

```
num1 = float(input("Enter first number: "))
```

```
num2 = float(input("Enter second number: "))
```

```
num3 = float(input("Enter third number: "))
```

```
num4 = float(input("Enter fourth number: "))
```

```
num5 = float(input("Enter fifth number: "))
```

```
sum = num1 + num2 + num3 + num4 + num5
```

```
print("sum:", sum)
```

8. Solve the following expressions:

a) $8+3*4-2**2$

b) $20\%7*3+10//3$

c) $2*3**2\%4+10-6/2$

Ans:

a) $8+3*4-2**2$

Step-by-step:

1. Exponentiation first: $2**2=4$

Expression becomes: $8+3*4-4$

2. **Multiplication next:** $3*4=12$

Expression becomes: $8+12-4$

3. **Addition and Subtraction** (from left to right):

$$8+12=20$$

$$20-4=16$$

Result: 16

b) $20\%7*3+10//3$

Step-by-step:

1. **Modulo and Multiplication** (same precedence, left to right):

First: $20\%7=6$ (remainder when 20 is divided by 7)

Expression becomes: $6*3+10//3$

2. **Multiplication:** $6*3=18$

Expression becomes: $18+10//3$

3. **Floor division:** $10//3=3$

Expression becomes: $18+3$

4. **Addition:** $18+3=21$

Result: 21

c) $2*3**2\%4+10-6/2$

Step-by-step:

1. **Exponentiation first:** $3**2=9$

Expression becomes: $2*9\%4+10-6/2$

2. **Multiplication, Modulo, Division** (same precedence, left to right):

o First: $2*9=18$

Expression becomes: $18\%4+10-6/2$

o Next: $18\%4=2$

Expression becomes: $2+10-6/2$

o Next: $6/2=3.0$ (float result)

Expression becomes: $2+10-3.0$

3. **Addition and Subtraction** (left to right):

$$2+10=12$$

$$12-3.0=9.0$$

Result: 9.0