

## LAB ACTIVITY 1

# INTRODUCTION TO PYTHON, PYTHON DATA TYPES-I

## Aim

Build a basic calculator program that allows the user to choose between addition, subtraction, multiplication, and division. Prompt the user to enter two numbers and select an operation. Perform the chosen operation and display the result.

## Theoretical Background

- **Python Syntax:** Python syntax refers to the rules governing how Python code should be structured. It includes conventions for indentation, use of keywords, and the order of statements.
- **Data Types:** Data types in Python specify the type of data that variables can hold. Common types include integers (whole numbers), floats (decimal numbers), and strings (text).
- **Variables:** Variables are used to store and manage data in Python. They act as containers for values and are accessed by variable names.
- **Input and Output Functions:** Python provides input and output functions. The 'input' function collects user input, while the 'print' function displays output.
- **Arithmetic Operators:** Python offers arithmetic operators like addition (+), subtraction (-), multiplication (\*), and division (/) for mathematical operations.
- **Conditional Statements:** Conditional statements (e.g., 'if', 'elif', 'else') make decisions in Python programs, allowing different code blocks to be executed based on specified conditions.
- **Comments:** Comments in Python are used for explanations and documentation. They provide insights into code purpose and functionality and are not executed by the program.

## Useful Python Commands/Functions

### 1. Input and Output Functions:

- `input(prompt)`: Reads a line from the console and returns it as a string.

```
1 user_input = input("Enter your name: ")
```

- `print(value)`: Prints the specified value to the console.

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

## 2. Variables and Data Types:

- `int(value)`, `float(value)`: Converts the given value to an integer or float, respectively.

```
1 num_as_string = "42"  
2 num_as_int = int(num_as_string)
```

- `str(value)`: Converts the given value to a string.

```
1 num_as_float = 3.14  
2 num_as_str = str(num_as_float)
```

- `type(variable)`: Returns the data type of the variable.

```
1 var_type = type(42)
```

## 3. Arithmetic Operators:

- `+` (addition)

```
1 sum_result = 10 + 5
```

- `-` (subtraction)

```
1 difference = 20 - 8
```

- `*` (multiplication)

```
1 product = 6 * 7
```

- `/` (division)

```
1 quotient = 15 / 3
```

- `**` (exponentiation)

```
1 result = 2 ** 3
```

## 4. Conditional Statements:

- `if condition:`

```
1 x = 10  
2 if x > 0:  
3     print("Positive")
```

- `elif condition:`

```
1 x = 0  
2 if x > 0:  
3     print("Positive")  
4 elif x == 0:  
5     print("Zero")
```

- `else:`

```
1 x = -5
2 if x > 0:
3     print("Positive")
4 elif x == 0:
5     print("Zero")
6 else:
7     print("Negative")
```

## 5. Math Functions (from the math module):

- `math.pi`: Returns the mathematical constant  $\pi$ .

```
1 import math
2 pi_value = math.pi
```

- `math.sqrt(x)`: Returns the square root of  $x$ .

```
1 import math
2 square_root = math.sqrt(16)
```

- `math.pow(x, y)`: Returns  $x$  raised to the power  $y$ .

```
1 import math
2 power_result = math.pow(2, 3)
```

## 6. String Manipulation:

- `len(string)`: Returns the length of the string.

```
1 text = "Hello, World!"
2 length = len(text)
```

- `string.lower()`, `string.upper()`: Converts the string to lowercase or uppercase, respectively.

```
1 text = "Hello, World!"
2 lower_case = text.lower()
3 upper_case = text.upper()
```

- `string.replace(old, new)`: Replaces occurrences of the old substring with the new substring.

```
1 text = "Hello, World!"
2 new_text = text.replace("World", "Python")
```

## 7. List Operations:

- `list.append(element)`: Appends the element to the end of the list.

```
1 numbers = [1, 2, 3]
2 numbers.append(4)
```

- `list.remove(element)`: Removes the first occurrence of the specified element from the list.

```
1 numbers = [1, 2, 3, 2]
2 numbers.remove(2)
```

- `list.sort()`: Sorts the elements of the list in ascending order.

```
1 numbers = [3, 1, 4, 1, 5, 9, 2]
2 numbers.sort()
```

## 8. Random Module (for the number-guessing game):

- `import random`: Imports the random module.

```
1 import random
```

- `random.randint(a, b)`: Returns a random integer between *a* and *b* (inclusive).

```
1 import random
2 random_number = random.randint(1, 100)
```

## 9. File Operations:

- `open(filename, mode)`: Opens a file in the specified mode ('r' for reading, 'w' for writing, 'a' for appending).

```
1 file = open("example.txt", "r")
```

- `file.read()`: Reads the contents of the file.

```
1 content = file.read()
```

- `file.readline()`: Reads a single line from the file.

```
1 line = file.readline()
```

## 10. Exception Handling:

- `try, except`: Used for handling exceptions and preventing program crashes.

```
1 try:
2     result = 10 / 0
3 except ZeroDivisionError:
4     print("Cannot divide by zero.")
```

These commands and functions cover a variety of tasks and can be combined creatively to solve the programming exercises in your chapter.

## Program

Write a Python program that acts as a basic calculator, allowing the user to input two numbers and choose an operation (addition, subtraction, multiplication, or division). Implement conditional statements to perform the selected operation and display the result, emphasizing proper syntax, variable usage, user input handling, and code documentation.

```
1 # Display a welcome message
2 print("Welcome to the Basic Calculator Program!")
3
4 # Prompt the user to enter two numbers
5 num1 = float(input("Enter the first number: "))
6 num2 = float(input("Enter the second number: "))
```

```
7
8 # Display a menu for operations
9 print("Select an operation:")
10 print("1. Addition (+)")
11 print("2. Subtraction (-)")
12 print("3. Multiplication (*)")
13 print("4. Division (/)")
14
15 # Prompt the user to choose an operation
16 operation = input("Enter the operation number (1-4): ")
17
18 # Implement conditional statements to perform the selected operation
19 if operation == '1':
20     result = num1 + num2
21     operator = '+'
22 elif operation == '2':
23     result = num1 - num2
24     operator = '-'
25 elif operation == '3':
26     result = num1 * num2
27     operator = '*'
28 elif operation == '4':
29     # Check if the second number is not zero for division
30     if num2 != 0:
31         result = num1 / num2
32         operator = '/'
33     else:
34         print("Error: Division by zero is not allowed.")
35         exit()
36 else:
37     print("Invalid operation. Please choose a number between 1 and 4.")
38     exit()
39
40 # Display the result
41 print(f"Result: {num1} {operator} {num2} = {result}")
```

## Output

```
1 Welcome to the Basic Calculator Program!
2 Enter the first number: 45
3 Enter the second number: 65
4 Select an operation:
5 1. Addition (+)
6 2. Subtraction (-)
7 3. Multiplication (*)
8 4. Division (/)
9 Enter the operation number (1-4): 1
10 Result: 45.0 + 65.0 = 110.0
```

```

1 Welcome to the Basic Calculator Program!
2 Enter the first number: 85
3 Enter the second number: 92
4 Select an operation:
5 1. Addition (+)
6 2. Subtraction (-)
7 3. Multiplication (*)
8 4. Division (/)
9 Enter the operation number (1-4): 3
10 Result: 85.0 * 92.0 = 7820.0

```

## Questions

1. Write a Python program that takes user input for their age, calculates the year they were born, checks if it's an odd or even year, and prints a message stating whether their age is an odd or even number, emphasizing the use of proper Python syntax, variables, input handling, arithmetic operations, conditional statements, and code documentation.

```

1 # Get user input for age
2 try:
3     age = int(input("Enter your age: "))
4 except ValueError:
5     print("Invalid input. Please enter a valid integer for age.")
6     exit()
7
8 # Calculate the birth year
9 current_year = 2024 # You can update this to the current year
10 birth_year = current_year - age
11
12 # Check if the birth year is odd or even
13 if birth_year % 2 == 0:
14     year_type = "even"
15 else:
16     year_type = "odd"
17
18 # Display the results
19 print(f"You were born in the year {birth_year}.")
20 print(f"Your age is {age} years, and the birth year is an {
    year_type} year.")

```

Output:

```

1 Enter your age: 21
2 You were born in the year 2003.
3 Your age is 21 years, and the birth year is an odd year.

```

2. Develop a Python script that calculates the area of a circle (given its radius), square (given its side), and rectangle (given both sides) and prints the result with an appropriate message.

```

1 import math
2
3 def calculate_circle_area(radius):

```

```

4     """Calculate the area of a circle."""
5     return math.pi * radius**2
6
7 def calculate_square_area(side):
8     """Calculate the area of a square."""
9     return side**2
10
11 def calculate_rectangle_area(length, width):
12     """Calculate the area of a rectangle."""
13     return length * width
14
15 # Get user input for the shape and dimensions
16 shape = input("Enter the shape (circle, square, or rectangle): ")
17     .lower()
18
19 if shape == "circle":
20     radius = float(input("Enter the radius of the circle: "))
21     area = calculate_circle_area(radius)
22     print(f"The area of the circle with radius {radius} is: {area
23           :.2f} square units.")
24 elif shape == "square":
25     side = float(input("Enter the side length of the square: "))
26     area = calculate_square_area(side)
27     print(f"The area of the square with side length {side} is: {
28           area:.2f} square units.")
29 elif shape == "rectangle":
30     length = float(input("Enter the length of the rectangle: "))
31     width = float(input("Enter the width of the rectangle: "))
32     area = calculate_rectangle_area(length, width)
33     print(f"The area of the rectangle with length {length} and
34           width {width} is: {area:.2f} square units.")
35 else:
36     print("Invalid shape. Please enter 'circle', 'square', or '
37           rectangle'.")

```

#### Output 1:

```

1 Enter the shape (circle, square, or rectangle): circle
2 Enter the radius of the circle: 5
3 The area of the circle with radius 5.0 is: 78.54 square units.

```

#### Output 2:

```

1 Enter the shape (circle, square, or rectangle): square
2 Enter the side length of the square: 5
3 The area of the square with side length 5.0 is: 25.00 square
  units.

```

#### Output 3:

```

1 Enter the shape (circle, square, or rectangle): rectangle
2 Enter the length of the rectangle: 5
3 Enter the width of the rectangle: 4
4 The area of the rectangle with length 5.0 and width 4.0 is: 20.00
  square units.

```

---

## Additional Python Programming Practice Questions

1. **Temperature Conversion:** Write a Python program that takes user input for a temperature in Celsius and converts it to Fahrenheit. Display the result with an appropriate message.
2. **Character Counter:** Develop a Python script that takes a user input string and counts the number of occurrences of each character in the string. Display the results in a readable format.
3. **Even or Odd Checker:** Create a Python program that prompts the user to enter an integer and checks if it's an even or odd number. Display a message indicating whether the number is even or odd.
4. **Quadratic Equation Solver:** Write a Python script that solves a quadratic equation of the form  $ax^2 + bx + c = 0$  given user input for coefficients  $a$ ,  $b$ , and  $c$ . Display the solutions with appropriate messages.
5. **Factorial Calculator:** Develop a Python program that calculates the factorial of a given non-negative integer entered by the user. Display the result with a suitable message.
6. **Guess the Number Game:** Create a simple number-guessing game in Python. Generate a random number between 1 and 100, and allow the user to guess the number. Provide hints (higher or lower) until the correct number is guessed.
7. **File Word Counter:** Write a Python script that reads a text file specified by the user and counts the number of words in the file. Display the word count with an appropriate message.
8. **List Operations:** Implement a Python program that performs the following operations on a list of numbers entered by the user:
  - Calculate the sum of the numbers.
  - Find the maximum and minimum values.
  - Sort the list in ascending order.
9. **Palindrome Checker:** Develop a Python program that checks if a given word entered by the user is a palindrome (reads the same backward as forward). Display the result with an appropriate message.
10. **Currency Converter:** Write a Python program that converts an amount in one currency to another. Allow the user to input the amount, original currency, and target currency. Display the converted amount.