4 Programming with Python

LEARNING OUTCOMES

After the lesson, students will be able to:

- Open and close Python.
- Take input.
- Work with the Input function.
- Use decision making statements.
- Create loops in a program.
- Use different loop control statements.
- Use keyboard shortcuts.

WARM UP

List any three real-life situations where a process is repeated several times.

Ans. Do it yourself.

CHAPTER NOTES

- Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.
- Python can be used:
  - On a server to create web applications
  - Alongside software to create workflows
  - To connect to database systems
  - To handle Big Data and perform complex mathematics
  - For software development

- While creating a program, we often need to interact with users, either to get data or to provide some sort of result. Most programs today use a dialog box as a way of asking some type of input from the user. Python provides us with inbuilt functions to read the input from the keyboard.
- The input( ) function first takes the input from the user and then evaluates the expression, which means Python automatically identifies whether the user entered a string or a number or list. If the input provided is not correct, then either syntax error or an exception is raised by Python.
- When input( ) function starts its execution, the program flow stops until the user has given an input.
- The text or message displayed on the output screen to ask a user to enter input value is optional, i.e., the prompt printed on the screen is optional.
- When you enter an input, the input() function converts it into a string. Even if you enter an integer value, the input( ) function converts it into a string. You need to convert it into an integer in your code using typecasting.
- Decision-making statements in programming languages decide the direction of flow of program execution. Decision-making statements available in Python are:
  - if statement
  - if - else statement
  - nested if statement
  - if - else - if statement
- The if statement is the most simple decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not, i.e., if a certain condition is true then a block of statements is executed, otherwise not.
- The if statement alone tells us that if a condition is true, it will execute a block of statements and if the condition is false, it won't. We can use the else statement with the if statement to execute a block of code when the condition is false.

- A nested if is an if statement that is the target of another if statement. Nested if statements mean an if statement inside another if statement. Python allows nesting of if statements within other if statements. That is, we can place an if statement inside another if statement.
- The if statement is executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the ladder is bypassed. If none of the conditions is true, then the final else statement will be executed.
- In Python, while loop is used to execute a block of statements repeatedly until a given condition is satisfied. And when the condition becomes false, the line immediately after the loop in the program is executed.
- For loops are used for sequential traversal. There is a for in loop which is similar to the for each loop in other languages.
- While using a while loop, if you forget to increment the counter variable in Python, or write flawed logic, the condition may never become false. In such a case, the loop will run infinitely, and the conditions after the loop will starve.
- Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.
- When you put a break statement in the body of a loop, the loop stops executing, and the control shifts to the first statement outside it. You can put it in a for or while loop.
- When program control reaches the continue statement, it skips the statements after 'continue'. It then shifts to the next item in the sequence and executes the block of code for it. You can use it with both the for and while loops.
- When we need a particular loop, class, or function in a program, but don't know what goes in it, we place the pass statement in it. It is a null statement. The interpreter does not ignore it but performs a no-operation (NOP).

DEMONSTRATION

- Opening and closing Python

- » Taking input
- » Working with the input function
- » Using decision-making statements
- » Creating loops in a program.
- » Using different loop control statements
- » Using keyboard shortcuts

LAB ACTIVITIES

1. Note what happens upon execution of the following:
   (a) Print n=4
   (b) Print 3+4
   (c) print 7.2, “this”, 9-5, “that”, 8/3.0
2. Calculate the following with the help of IDLE:
   (a) 7+8*12
   (b) (7+8)*12
3. Find out what happens when you type these mathematical explanations:
   (a) 7**1000
   (b) 1/0
4. Find out the end result:
   (a) a = 2- 7 + 14
   (b) b = 3*9
   (c) c = 6.0/5.0
   print “These are the values:”, a, b, c

ASSESSMENT

Teacher can assess the students with an oral quiz on the input() function, decision-making statements, loops and loop control statements.