

# 6. CONTROL STRUCTURES

## Learning Objectives

After studying this lesson, students will be able to:

- To gain knowledge on the various flow of control in Python language.
- To learn through the syntax how to use conditional construct to improve the efficiency of the program flow.
- To apply iteration structures to develop code to repeat the program segment for specific number of times or till the condition is satisfied.

## Important Notes and Points

- ❖ A program statement that causes a jump of control from one part of the program to another is called **control structure** or **control statement**.
- ❖ A **sequential statement** is composed of a sequence of statements which are executed one after another.
- ❖ Decision making is what we are to learn through alternative or branching statement.
- ❖ The condition specified in the if is checked, if it is true, the value of variable1 is stored in variable on the left side of the assignment, otherwise variable2 is taken as the value.

- ❖ **if..elif..else** statement is similar to nested if statement
- ❖ The two blocks of code in our example of if-statement are both **indented** four spaces, which is a typical amount of **indentation** for **Python**. In most other programming languages, **indentation** is used only to help make the code look pretty. But in **Python**, it is required to indicate to which block of code the statement belongs to.
- ❖ That the control variable is **i**, which is initialized to 10, the condition is tested **i<=15**, if true value of **i** gets printed, then the control variable **i** gets updated as **i=i+1** (this can also be written as **i +=1** using shorthand assignment operator). When **i** becomes **16**, the condition is tested False and this will terminate the loop.
- ❖ **print** can have *end*, *sep* as parameters. *end* parameter can be used when we need to give any escape sequences like `'\t'` for tab, `'\n'` for new line and so on. *sep* as parameter can be used to specify any special characters like, (comma) ; (semicolon) as separator between values (Recall the concept which you have learnt in previous chapter about the formatting options in `print()`).
- ❖ In Python, indentation is important in loop and other control statements. Indentation only creates blocks and sub-blocks like how we create blocks within a set of { } in languages like C, C++ etc.

- ❖ range () can also take values from string, list, dictionary etc. which will be dealt in the later chapters.
- ❖ The jump statement in Python, is used to unconditionally transfer the control from one part of the program to another. There are three keywords to achieve jump statements in Python : **break, continue, pass.**
- ❖ The **break** statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop.
- ❖ Continue statement unlike the break statement is used to skip the remaining part of a loop and start with next iteration.
- ❖ **pass** statement is generally used as a placeholder. When we have a loop or function that is to be implemented in the future and not now, we cannot develop such functions or loops with empty body segment because the interpreter would raise an error. So, to avoid this we can use pass statement to construct a body that does nothing.