UNIT – I: COMPUTATIONAL THINKING AND PROGRAMMING - 2

CHAPTER-1

REVISION OF THE BASICS OF PYTHON

Topic-1 Python Basics

Concepts Covered
- *Python Programming Basics*
- *it's features,*
- *tokens,*
- *data types,*
- *operators, etc.*

Revision Notes

- Python was created by Guido Van Rossum in late '1980s' (Released in 1991) at National Research Institute in the Netherland. Python got its name from a BBC comedy series – "Monty Python's Flying Circus". Python is based on two programming languages:

  (i) ABC language

  (ii) Modula 3

Some qualities of Python based on the programming fundamentals are given below:

- **Interactive Mode:** Interactive mode, as the name suggests, allows us to work interactively. It executes the code by typing one command at a time in Python shell.
- **Script Mode:** In script mode, we type Python program in a file and then execute the content of the file.
- **Indentation:** Blocks of code are denoted by line indentation, which is rigidly enforced.
- **Comments:** A hash sign (#) that is not inside a string literal begins a single line comment. We can use triple quoted string for giving multi-line comments.
- **Variables:** A variable in Python is defined through assignment. There is no concept of declaring a variable Other than assignment. Value of variable can be manipulated during program run.
- **Dynamic Typing:** In Python, while the value that a variable points to has a type, the variable itself has no strict type in its definition.
- **Static Typing:** In static typing, a data type is attached with a variable when it is defined first and it is fixed.
- **Multiple Assignment:** Python allows you to assign a single value to several variables and multiple values to multiple variables simultaneously.

  For example:
  ```
  a = b = c = 1
  a, b, c = 1, 2, "john"
  ```
- **Token:** The smallest individual unit in a program is known as a Token or a lexical unit.
- **Identifiers:** An identifier is a name used to identify a variable, function, class, module, or other object. An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores, and digits (0 to 9).

  Python does not allow punctuation characters such as @, $, and % within identifiers. Python is a **case sensitive programming** language. Thus, Value and value are two different identifiers in Python.

Here are identifiers naming convention for Python:

- Class names start with an uppercase letter and all other identifiers with a lowercase letter.
- Starting an identifier with a single leading underscore indicates by convention that the identifier is meant to be private.
- Starting an identifier with two leading underscores indicates a strongly private identifier.
- If the identifier ends with two trailing underscores, the identifier is a language-defined special name.
- **Words (Keywords):** The following list shows the **reserved words** in Python v3.0 or later.

Key Words

Case Sensitive Programming: Case sensitivity describes a programming language's (C, C++, Java, C#, Verilog, Ruby, Python and Swift) ability to distinguish between upper and lower case versions of a letter.

Reserved Words: Reserved words are all those words that have a strictly predefined meaning—they are reserved for special use and cannot be used for any other purpose.

Maps or Mapping: A mapping type is a data type comprised of a collection of keys and associated values. Python's only built-in mapping type is the dictionary.

Integer & Long: To store whole numbers i.e. decimal digits without fraction part.

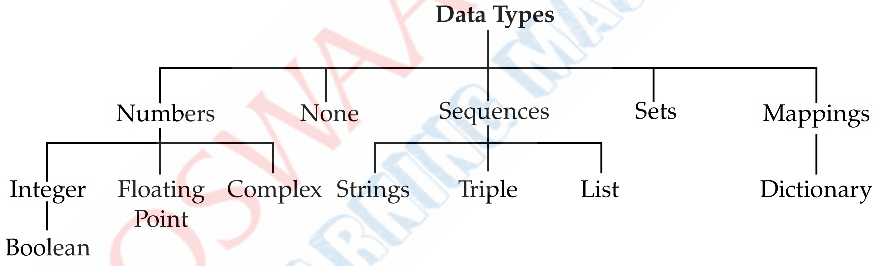
Float/floating point: To store numbers with fraction part.

Complex: To store real and imaginary part.

Python Keyword List

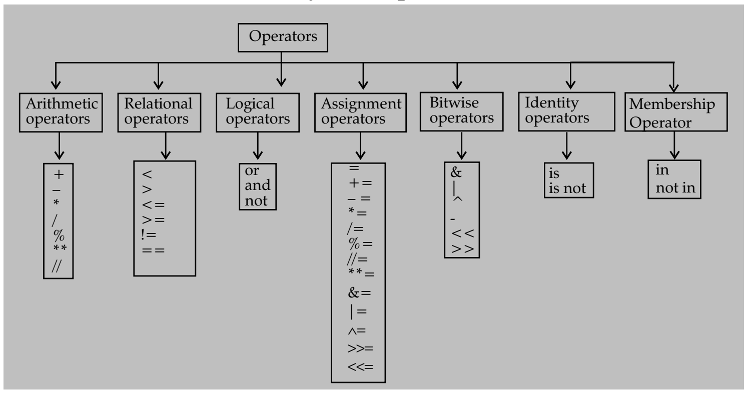
False	None	True	and	as	assert	async (v3.5 or later)
await (v3.5 or later)	break	class	continue	def	del	elif
else	except	finally	for	from	global	if
import	in	is	lambda	nonlocal	not	or
pass	raise	return	try	while	with	yield

- These reserved words should not be used as constant or variable or any other identifier names. All the Python keywords contain lowercase letters only except False, None and True which have first letter capital.
- **Literal/Constant:** Literals (Often referred to as constant value) are data items that have a fixed value. Python allows several kind of literals as String literals, Numeric literals, Boolean literals, special literal None and literal Collections
- **Data Types:** Data type is a set of values and the allowable operations on those values. Python has a great set of useful data types. Python's data types are built in the core of the language. They are easy to use and straight forward.



- Numbers can be either **integer/floating point** or **complex** type numbers.
- **None:** It is a special data type with a single value. It is used to signify the absence of value in a situation.
- A sequence is an ordered collection of items, indexed by integers starting from 0. Sequences can be grouped into strings, tuples and lists.
  - Strings are lines of text that can contain any character. They can be declared with double quotes.
  - Lists are used to group other data. They are similar to arrays.
  - A **tuple** consists of a number of values separated by commas.
- A **set** is an unordered collection with no **duplicate** items.
- A **dictionary** is an unordered set of key value pairs where the keys are unique.
- **Operator:** Operators are special symbols which perform some computation. Operators and operands form an expression. Python operators can be classified as given below :

Python Operators



► **Expressions:** An expression in Python is any valid combination of operators, literals and variables.

Topic-2

Conditional and Iterative Statements

Concepts Covered

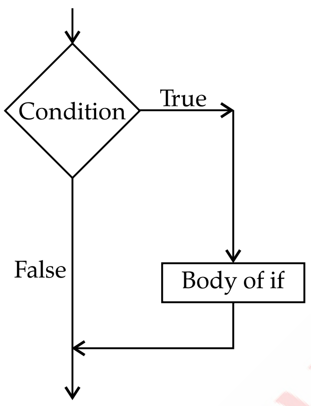
- *If*
- *If-else*
- *If-elif-else*
- *Nested if*
- *Programs based on above statements*
- *Concept Covered for loop*
- *while loop*
- *range function*
- *break and continue function*
- *nested loops*
- *programs based on above loops and statements*

Revision Notes

► A conditional is a statement which is executed, on the basis of result of a condition.

- **if conditionals statement in Python have the following forms.**

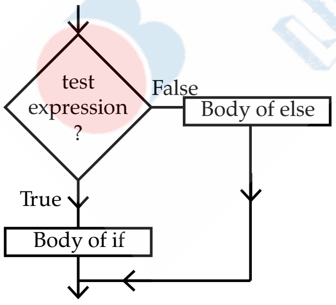
(A) Simple if



Syntax:

```
if < conditional expression>:
    [statements]
```

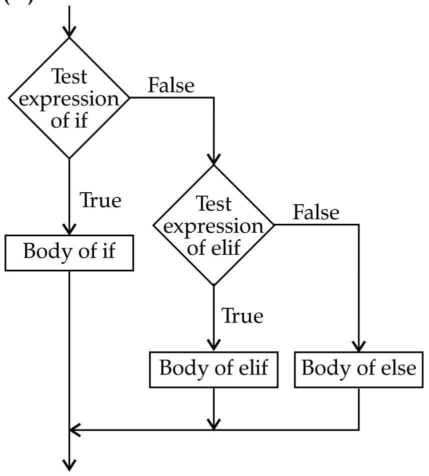
(B) The if-else conditional



Syntax:

```
if <conditional expression>:
    [statements]
else:
    [statements]
```

(C) The if-elif-else conditional statement



Syntax:

```
if <conditional expression>:
    Statement
        [statements]
elif <conditional expression>:
    statement
    [statements]
else:
    statement
    [statements]
```

(D) Nested if

- A nested if is an if which is inside another if's body or elif's body or else's body.

Syntax:

The syntax of the nested if...elif...else construct may be:

```
if expression1:
    statement(s)
    if expression2:
        statement(s)
    elif expression3:
        statement(s)
    elif expression4:
        statement(s)
    else:
        statement(s)
else:
    statement(s)
```

- Storing conditions – Complex and repetitive conditions can be named and then used in if statements.

► The iteration statements or repetition statements allow a set of instructions to be performed repeatedly.

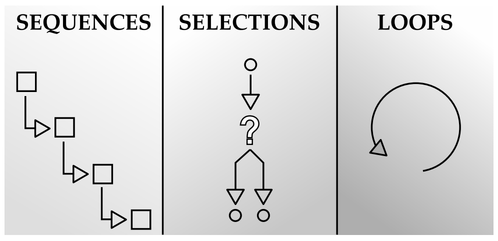
► Python provides three types of loops

(A) Counting loops repeat a certain number of times e.g. for

(B) Conditional loops repeat until a certain condition is true e.g. while.

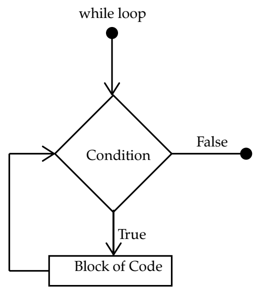
(C) Nested loops.

Key Diagram



1. **Python While Loop**

A while loop in Python iterates till its condition becomes False. In other words, it executes the block of statements until the condition it takes is true.



Syntax

```
while <logical Expression>:
    loop body
```

When the program control reaches the while loop, the condition is checked. If the condition is true, the block of code under it is executed. After that, the condition is checked again. This continues until the condition becomes false. Then the first statement, if any after the loop is executed. Remember to indent all statements under the loop equally.

e.g.

```
>>> a=3
>>> while(a>0):
    print(a)
    a-=1
```

Output

```
    3
    2
    1
```

(a) An Infinite Loop

Be careful while using a while loop. Because if you forget to increment or decrement 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. To stop execution, press Ctrl+C. However, an infinite loop may sometimes be useful.

(b) The else statement for while loop

A while loop may have an else statement after it. When the condition becomes false, the block under the else statement is executed. However, it doesn't execute if you break out of the loop or if an exception is raised.

e.g.

```
>>> a=3
>>> while(a>0):
            print(a)
            a-=1
    else:
        print("Reached 0")
```

Output

```
    3
    2
    1
    Reached ( )
```

In the following code, we put a break statement in the body of the while loop for a==1. So, when that happens, the statement in the else block is not executed.

e.g.

```
>>> a=3
>>> while(a>0):
    print(a)
    a-=1
    if(a==1):
        break
    else:
        print("Reached 0")
```

Output

```
3
2
```

(c) Single Statement while

Like an if statement, if we have only one statement in while loop's body, we can write it all in one line.

e.g.

```
>>> a=3
>>> while a>0: print(a); a-=1;
```

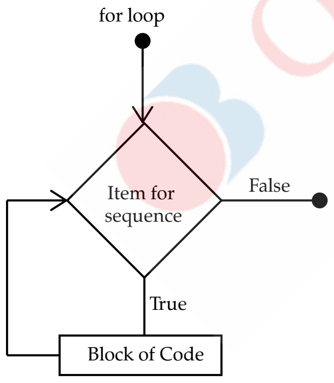
Output

```
3
2
1
```

You can see that there were two statements in while loop's body, but we used semicolons to separate them. Without the second statement, it would form an infinite loop.

2. Python for Loop

Python for loop can iterate over a sequence of items. The structure of a for loop in Python is different than that in C++ or Java. That is, for(int i=0;i<n;i++) won't work here. In Python, we use the 'in' keyword.



```
>>> for a in range(3):
    print(a)
```

Output

```
0
1
2
```

If we wanted to print 1 to 3, we could write the following code.

```
>>> for a in range(3):
    print(a+1)
```

Output

```
1
2
3
```

(a) The range() function

This function yields a sequence of numbers. When called with one argument, say n, it creates a sequence of numbers from 0 to n-1.

```
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

We use the list function to convert the range object into a list object. Calling it with two arguments creates a sequence of numbers from first to second.

```
>>> list(range(2,7))
[2, 3, 4, 5, 6]
```

You can also pass three arguments. The third argument is the interval.

```
>>> list(range(2,12,2))
[2, 4, 6, 8, 10]
```

Remember, the interval can also be negative.

```
>>> list(range(12,2,-2))
[12, 10, 8, 6, 4]
```

3. Nested Loops

A loop may contain another loop in its body, this inner loop is called nested loop. But in a nested loop, the inner loop must terminate before the outer loop.

e.g.

```
for i in range(1,6):
    for j in range (1,i):
        print("*", end=" ")
    print()
```

- **Jump Statements:** Python offers two jump statements-break and continue to be used within loops to jump out of loop iterations.
  - **break statement:** It terminates the loop it lies within. It skips the rest of the loop and jumps over to the statement following the loop.
  - **continue statement:** Unlike break statement, the continue statement forces the next iteration of the loop to take place, skipping any code in between.

Key Words

Indentation: In most programming languages, the statements within a block are put inside curly brackets. However, Python uses indentation for block as well as for nested block structures. Leading whitespace (spaces and tabs) at the beginning of a statement is called indentation.

Control Structure: Control Structure is a programming language construct which affects the flow of the execution of program.

Topic-3 Concept of Debugging and Python's Modules

Concepts Covered
- *Syntax Errors*
- *Logical Errors*
- *Runtime Errors*
- *Modules*
- *import and from statements*
- *math module*
- *random module*
- *statistics module*

Revision Notes

- An error or a bug is anything in the code that prevents a program from compiling and running correctly.
- There are three types of errors:

Compile Time errors occur at compile time.

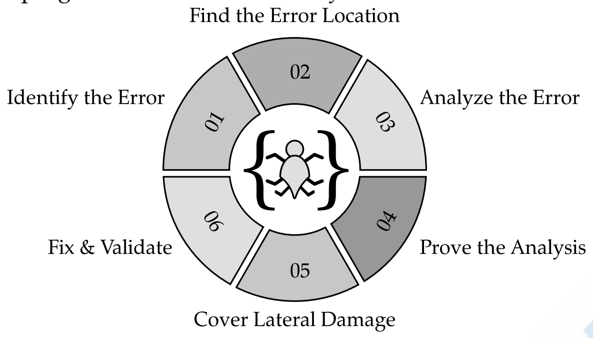
These are of two types :

(i) Syntax errors occur when rules of programming language are misused.

(ii) Semantics errors occur when statements are not meaningful.

Run Time errors occur during the execution of a program.

Logical Errors occur due to programmer's mistaken analysis of the error.



To remove logical errors is called **debugging**.

- A Python module can contain objects like docstrings, variables constants, classes, objects, statements, functions.
- Modules are accessed by using the import statement. A module is loaded only once, regardless the number of times it is imported.

  Syntax

  (i) import module_name

  (ii) from <module> import <object>
- **Mathematical functions:** Mathematical operations can be performed by importing the math module. Different types of mathematical functions:

  (i) sqrt(): find the square root of a specified expression

  (ii) pow(): compute the power of a number

  (iii) fabs(): Returns the absolute value of x.

  (iv) ceil(x): returns smallest integer value greater than or equal to x.

  (v) floor(x): returns the largest integer value less than or equal to x.
- **Random Functions:** Python offers random module that can generate random numbers. Different random functions are as follows

  (i) random(): Used to generate a float random number less than 1 and greater than or equal to 0.

  (ii) choice(): Used to generate 1 random number from a container.
- **Statistics Module:** To access Python's statistics functions, we need to import the functions from the statistics module.

  Some statistics functions are as follows:

  (i) mean(): Returns the simple arithmetic mean of data which can be a sequence or an iterator.

  (ii) median(): Calculates middle value of the arithmetic data in iterative order.

  (iii) mode() : Returns the number with maximum number of occurrences.

Key Words

Debugging: Debugging is the process of detecting and removing of existing and potential errors in a software code that can cause it to behave unexpectedly or crash.

Debugger: Debugger is a computer program used by programmers to test and debug a target program.

Topic-4 List, Tuples and Dictionary

Concepts Covered

- *List, Tuple and Dictionary data types*
- *Operations performed on these data types*

Revision Notes

- **List**
  - A list is a standard data type of Python that can store a sequence of values belonging to any type.
  - The lists are depicted through square brackets.

- These are mutable (i.e. modifiable), you can change elements of a list in place.
- Lists store a reference at each index.
- We can index, slice and access individual list elements.
- len (L) returns the number of items in the list L.Membership operators in and not in can be used with list.
- To join two lists, use `+' (concatenation) operator.
- L [start: stop] creates a list slice with starting index as start till stop as stopping index but excluding stop.
- List manipulation functions are
  append(), insert(), extend(),sort(), remove(), reverse() and pop().

► **Tuples**

- Tuples are immutable Python sequences, *i.e.* you cannot change elements of a tuple in place.
- Tuples' items are indexed.
- Tuples store a reference at each index.
- Tuples can be indexed sliced and its individual items can be indexed.
- len (T) returns count of tuple elements.
- Tuple manipulation functions are: len(), max(), min(), and tuple().

► **Dictionaries**

- Dictionaries in Python are a collection of some key-value pairs.
- These are mutable and unordered collection with elements in the form of a key : value pairs that associate keys to values.
- The keys of dictionaries are immutable type and unique.
- To manipulate dictionaries functions are: len(), clear(), has_key(),items(), keys(), values(), update().
- The membership operators in and not in work with dictionary keys only.

Key Words

Slicing: In Python it is a feature that enables accessing parts of sequences like strings, tuples and lists.

Packing: In Python, tuples are collections of elements which are separated by commas. It packs elements or value together so, this is called packing.

► **Python Data Types**

String	List	Tuple	Set	Dictionary
immutable	Mutable	Immutable	Mutable	Mutable
Ordered/Indexed	Ordered/Indexed	Ordered/Indexed	Unordered	Unordered
Allows duplicate values	Allows duplicate values	Allows duplicate values	Does not allow duplicate values	Does not allow duplicate keys
Empty string=""	Empty list=[]	Empty tuple=()	Empty set=set()	Empty dictionary= {}
String with single element="A"	List with single item=["Apple"]	Tuple with single item=("Apple",)	Set with single item={"Apple"}	Dictionary with single item={"Hello":1}
Stores characters or strings	It can store any data type str, list, set, tuple, int and dictionary	It can store any data type str, list, set, tuple, int and dictionary	It can store data types (int, str, tuple) but not (list, set, dictionary)	Inside of dictionary key can be int, str and tuple only values can be of any data type int, str, list, tuple, set and dictionary.

Topic-5 Strings in Python

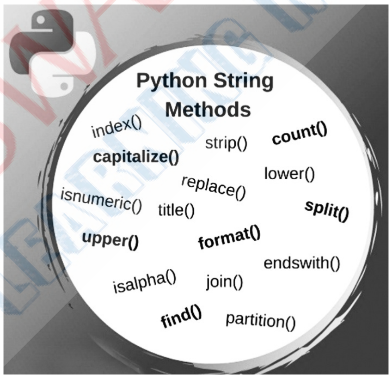
Concepts Covered

- *Strings*
- *String Operations like:*
  Concatenation *Repetition*
  Membership *Slicing*
  Traversing *Built-in Functions of Strings*

Revision Notes

- Strings in Python are stored as individual character in contiguous location, with two way index for each location.
- Strings are immutable and hence item assignment is not supported.
- Following operations can be used on strings.
  (1) Concatenation `+'
  (2) Replication `*'
  (3) Membership Operators as in and not in
  (4) Comparison Operators as = =, ! =, <, >, < =, > =
- Built in functions
  ord() – returns ASCII value of passed character.
  chr() – returns character corresponding to passed ASCII code
- String slice refers to a part of the string, where strings are sliced using a range of indices
  Syntax : string [n:m].

Key Diagram



CHAPTER-2 FUNCTIONS

Revision Notes

- A **function** is a named block of statements that can be invoked by its name. A function is organized and reusable code that is used to perform a single, given action. Functions provide better modularity for your application and a high degree of code reusability.
  Advantages of functions:
  - Program development made easy and fast

- Program testing becomes easy
- Code sharing becomes possible
- Code re-usability increases
- Increases program readability

► Function blocks begin with the keyword def followed by the function name and parentheses e.g def sum( ): Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses. The first statement of a function can be an optional statement - the documentation string of the function or docstring, The code block within every function starts with a colon (:) and is indented. The statement return [expression] exit a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None. Defining a function only gives a name, specifies the parameters that are to be included in the function, a structure the blocks of code.

► **Example:** A simple function that prints the message "Hello Python".

```
#function definition
def hello_Python():
   print("hello python")
# function calling
hello_Python()
```

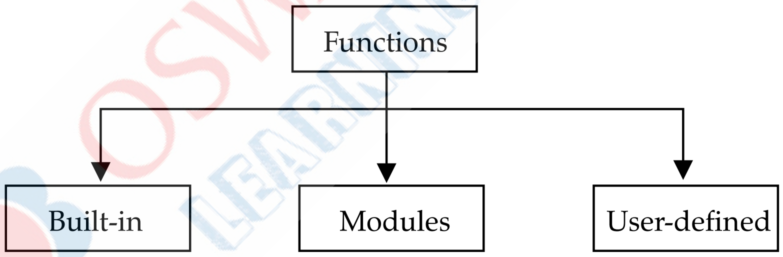
Output:

```
hello Python
```

Advantages of functions:

- Program development made easy and fast
- Program testing becomes easy
- Code sharing becomes possible
- Code re-usability increases
- Increases program readability

Types of functions:



- The built-in functions of Python are always available, one needs not import required module for them.

► **Examples of Some Built-in Functions**

(i) **print():** It prints objects to the text stream file.
(ii) **input():** It reads the input, converts it to a string and returns that.
(iii) **sorted():** Returns a new sorted list from the items in iterable.
(iv) **bool():** Returns a boolean value i.e., True or False.
(v) **min():** Returns the smallest of two or more arguments.
(vi) **any():** Returns True if any element of the iterable is True.

Modules: A module is a file containing Python definitions and statements. A Python module has .py extension. The Python modules that come preloaded with Python are called "Standard library modules". Python modules can be imported in a program using import statement. The math module of Python provides mathematical functionality.

- exp(x): Return e**x
- log(x,(base)): With one argument, returns the natural logarithm of x (to base e). With two arguments, returns logarithm of x to the given base calculate as log(x)/ log(base)
- log10(x): Returns logarithm of x at base 10. This is usually more accurate than log(x,10).

- pow(x, y): Returns x raised to the power y. In particular, pow(1.0, x) and pow(x, 0.0) always return 1.0, even when x is a zero or a NaN. If both x and y are finite, x is negative, and y is not an integer then pow(x, y) is undefined, and raises ValueError.
- sqrt(x): Returns the square root of x.
- cos(x): Returns the cosine of x radians.
- sin(x): Returns the sine of x radians.
- tan(x): Returns the tangent of x radians.
- degrees(x): Converts angle x from radians to degrees.
- radians(x): Converts angle x from degrees to radians.
- **ceil(x):** Returns the ceiling of x as a float, the smallest integer value greater than or equal to x.
- **floor(x):** Returns floor of x as a float, the largest integer value less than or equal to x.
- **fabs(x):** Returns the floating point absolute value of x.
- String Functions
  - **(i) partition():** It splits the string at the first occurrence of the given argument and returns a tuple containing three parts.
  - **(ii) join():** It takes a list of string and joins them as a regular string.
  - **(iii) split():** It splits the whole string into the items with separator as a delimiter.

► **User-Defined Functions:** User defined functions are those that we define ourselves in our program and then call them wherever we need.

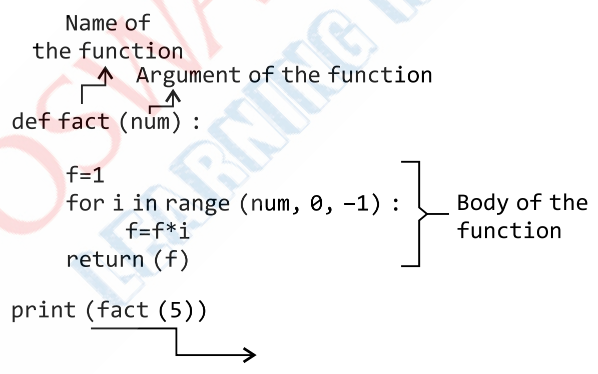
► Syntax to create USER DEFINED FUNCTION

def function_name([comma separated list of parameters]):

statements….

statements….

Example



```
def fact (num) :

    f=1
    for i in range (num, 0, -1) :
        f=f*i
    return (f)

print (fact (5))
```

Points to remember:

- Keyword def marks the start of function header
- Function name must be unique and follows naming rules same as for identifiers
- Function can take arguments. It is optional
- A colon(:) marks the end of function header
- Function can contain one or more statements to perform specific task
- Function must be called/invoked to execute its code

► The scope of a variable determines the portion of the program where you can access a particular identifier. There are two basic scopes of variables in Python:

(i) Global variables

(ii) Local variables

Variables that are defined inside a function body have a local scope, and those defined outside have a global scope. All variables in a program may not be accessible at all locations in that program. This depends on where you have declared a variable or when it is called the scope of variable.

► Consider the following python program: Add two numbers using function

```
Local      1. def calcSum(x, y):
Variables  2.       z=x+y                     #Definition of function calcSum
           3.       return z

             #Main Program
Global     4.  Num1 = int(input("Enter First Number :"))
Variables  5.  Num2 = int(input("Enter Second Number :"))
           6.  Sum = calcSum (Num1, Num2)          #Calling of function
           7.  Print ("Sum of given Numbers =", Sum)
```

► **Passing different objects as arguments**

You can send any data types of argument to a function as string, number, list, dictionary etc., and it will be treated as the same data type inside a function.

e.g. List as an argument

```
    def fun(Fruit):
        for i in Fruit:
        print(i)
  Food = ["Mango", "Cherry", "Grapes", "Banana"]
  fun(Food)
```

Output:

```
Mango
Cherry
Grapes
Banana
```

- **Arguments:** The values being passed through a function call are called arguments or actual parameters.
- **Parameters:** The Values being received in the function definition are called parameters or formal parameters.

  Consider the following example,

```
                                 a, b are Parameters / Formal Parameters
def multiply (a, b):
    print (a * b)                #Definition of function multiply
# Main Program
y = 3
multiply (12, y)                 #Function Call 1
x = 5                     Value 12 and variable y are Arguments/Actual Parameters
multiply (y, x)                  #Function Call 2
                          Variables y and x are Arguments/Actual Parameters
```

► **Parameter Passing**

Python supports three types of parameters/ formal parameters:

(i) **Positional (or Required) Parameters:** When the function call statement matches the number and order of arguments as defined in the function definition, is called the positional parameter.

(ii) **Default Parameters:** A parameter having default value in the function header so that the value can be used if the corresponding actual parameter is missing is known as a default parameter.

(iii) **Keyword (or Named) Parameters:** The named arguments with assigned values being passed in the function call statement.

(iv) **Variable-length Parameters:** Pass multiple values with single argument name.

► **Return Values**

Python functions may or may not return a value. There are broadly two types of functions in python based on return values:

(i) Functions returning some values (non-void functions): These functions return some computed result in terms of a value to the caller.

Syntax:

```
return <value>
```

(ii) Functions not returning any values (void functions): These functions do not return any computed or final value to the caller.

A void function may or may not have a return statement. If a void function has a return statement, then it takes the following syntax.

Syntax:

```
return
```

(iii) Returning Multiple values from python Functions: Python let you return more than one value from a function.

Syntax:

```
return <value1>, <value2>, <value3>, . . .
```

► **Mutable/Immutable Properties of Data Objects**

- Everything in Python is an object and every objects in Python can be either mutable or immutable.
- In Python, immutable objects are those whose value cannot be changed in place after assignment or initialization. They allocate new memory whenever their value is changed. Examples of immutable data types or objects or variables in Python are numbers (integers, floats), strings and tuples.
- In Python, list, dict and set are examples of mutable data types. Their values can be changed in place after their assignment or creation. When the value of a mutable variable is changed its memory is not reallocated.

The value of integer, float, string or tuple is not changed in the calling block if their value is changed inside the function or method block but the value of list, dict or set object is changed.

Python immutable objects, such as numbers, tuple and strings, are also passed by reference like mutable objects, such as list, set and dict.

► **Passing Strings, Lists, Tuples, Dictionaries to Functions**

You can send any data types of argument to a function (string, number, list, dictionary etc.), and it will be treated as the same data type inside the function.

E.g. if you send a List as an argument, it will still be a List when it reaches the function.

String can be passed in a function as argument as **pass by value.**

Key Words

Call by Value: In call by value (also known as pass by value), a parameter acts within the function as a new local variable initialized to the value of the argument (a local (isolated) copy of the argument).

In **call by reference** (also known as pass by reference), the argument variable supplied by the caller can be affected by actions within the called function

Unpacking: is the process of getting out stuff — iterables such as lists, tuples, and dictionaries.

In Python, you can unpack list, tuple, dict (dictionary) and pass its elements to function as arguments by adding * to list or tuple and ** to dictionary when calling function.

Now we want to pass a list that contains multiple elements and these elements act as multiple arguments of a function. This concept of passing a single list of elements as multiple arguments is known as **Unpacking Argument List**.

We use *args to unpack the single argument into multiple arguments. We use the unpacking operator * when arguments are not available separately.

In function call, we have to explicitly define/pass the tuple. It is not required to specify the data type as tuple in formal argument. Usually, they can be used to pass to functions and can have different kinds of behavior. Different cases can arise.

Case 1: fnc(a, b): Sends a and b as separate elements to fnc.

Case 2: fnc((a, b)): Sends (a, b), whole tuple as 1 single entity, one element.

Case 3: fnc(*(a, b)): Sends both, a and b as in Case 1, as separate integers.

► In Python, a number of mathematical operations can be performed with ease by importing a module named "math" which defines various functions which makes our task easier.

► **Flow of Execution:** Flow of execution can be defined as the order in which the statements in a program are executed. The Python interpreter starts executing the instructions in a program from the first statement. The statements are executed one by one, in the order of appearance from top to bottom.

► If a def statement is encountered all the statements of the function are skipped but the function header is interpreted to check if it is valid.

► If a function call is encountered the statements in the called function are executed from top to bottom.

Key Words

- **Global Variables** are the one that are defined and declared outside a function and we need to use them inside a function.
- **Local Variables:** A variable declared inside the function's body and in the local scope is known as a local variable.
- **Doc Strings** are triple quoted string in Python module program which are displayed as document when help command is used.
- **Modularity:** The act of partitioning a program into individual components (modules) is called modularity.
- **Parameters** are variables listed within parentheses of a function header.

CHAPTER-3

FILE HANDLING

Revision Notes

- **Files** are used to store huge collection of data and records permanently.
- Many applications require large amount of data. In such situation, we need to use some devices such as hard disk, compact disc etc., to store the data.
- **Need for a data file**
  - It is a convenient way to deal with large quantities of data.
  - To avoid input of data multiple times during program execution.
  - To share data between various programs.
- **Types of file:**
  - **Text files:** Text files store information in ASCII or Unicode characters. In text file, each line of text is terminated, (delimited) with a special character known as EOL (End of Line) character.
  - **Binary files:** Binary files are just files that contain information in the same format in which the information is held in memory, i.e., in binary file, there is no delimiter for a line.
  - **CSV files:** CSV (Comma Separated Value) files are a common file format for transferring and storing data. CSV is a widely used format that stores tabular data (numbers and text) as plain text. It is a delimited text file that uses a comma to separate values.
- **In Python, File Handling consists of following three steps:**
  - Open the file.
  - Process file i.e perform read or write operation.
  - Close the file.
- **File Manipulation**

  One of the key features of most object oriented programming languages is their ability to manipulate files. This involves creating files, reading from them, opening them, writing to them, and appending text (or data) to them.
- **Basic Operations on a Text File**
  - Open a text file: Files in Python can be opened with a built-in open() function. The open() function creates a file object which would be utilized to call other methods associated with it.

    Syntax:

    file_object=open(filename[access_mode],[buffering])

    Here is the parameter details:
  - **filename:** The file name argument is a string value that contains the name of the file that you want to access.
  - **access_mode:** The access_mode determines the mode in which the file has to be opened i.e., read, write, append, etc. A complete list of possible values is given below in the table. This is optional parameter and the default file access mode is read (r).

File Opening Modes

MODES	DESCRIPTION
r	Opens a file for reading only in text format. The file pointer is placed at the beginning of the file. This is the default mode.
rb	Opens a file for reading only in text format. The file pointer is placed at the beginning of the file. This is the default mode.
r+	Opens a file for both reading and writing. The file pointer will be at the beginning of the file.
rb+	Opens a file for both reading and writing in binary format. The file pointer will be at the beginning of the file.
w	Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
wb	Opens a file for writing in binary format. Overwrites the file if the file exists. If the file does not exist, creates a new file for reading and writing.
w+	Opens a file for reading as well as writing. Overwrites the file if it already exists and creates a new file if it doesn't exist.
wb+	Opens a file for writing as well as reading in binary mode. It will overwrite an already existing file and create a new file if it doesn't exist. The file pointer is placed at the beginning of the file.
a	Opens a file for appending. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing.
ab	Opens a file for appending in binary format. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing.
a+	Opens a file for both appending and reading. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing.
ab+	Opens a file for both appending and reading in binary format. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing.

- **Buffering:** If the buffering value is set to 0, no buffering will take place. If the buffering value is 1, line buffering will be performed while accessing a file. If you specify the buffering value as an integer greater than 1, then buffering action will be performed with the indicated buffer size. If negative, the buffer size is the system default (default behaviour).
- **Close a text file:** after opening and performing the reading, writing operations, it is important to close the file. This is done using .close() method.

  Syntax:

  file.close()

Reading and Manipulation of data from a text file

Before start reading the file, we must open the file in 'r' mode.

There are three ways in which we can read the files in python.

- read([n]): Read the entire file into a string.

  Syntax:

  file. read([n])
- readline([n]): Read next line (upto and include newline) and return a string (including newline). It returns an empty string after the end-of-file (EOF).

  Syntax:

  file. readlines([n])
- readlines():Read all lines into a list of strings.

  Syntax:

  file. readlines([n])

  Here, n is the number of bytes to be read.

  In order to write data into a file, we must open the file in write mode. We have two methods for writing data into a file as shown below.

- **write(string):** Write the given string to the file and return the number of characters written. You need to explicitly terminate the string with a '\n', if needed.

  Syntax:

  ```
  file.write(string)
  ```
- writelines(list): Using this function we can give list of lines to write into the file. The important thing to remember here is you have to give '\n' manually at the end of each line in the list to tell the python to write it into separate lines.

  Eg. ['1st Line \n', '2nd Line \n', '3rd Line']

  Syntax:

  file. writelines(list)

Appending data into a text file

To append data into a file we must open the file in 'a+' mode so that we will have access to both the append as well as write modes.

For example, consider we already have a file "text1.txt" with data written "I am in class-XII.". Let's perform append operation on that file.

```
f.open("text1.txt", 'a')
f.write("I read Oswaal's books.")
f.close()
```

Output:

```
# I am in class-XII. I read Oswaal's books.
```

Standard Input / Output and Error Streams

Python's sys module provides us with file-like objects that represent stdin, stdout, and stderr.

stdin , stdout , and stdderr variables contain stream objects corresponding to standard I/O streams.

- **Standard input:** This is the file-handle that a user program reads to get information from the user. We give input to the standard input (stdin).
- **Standard output:** The user program writes normal information to this file-handle. The output is returned via the Standard output (stdout).
- **Standard error:** The user program writes error information to this file-handle. Errors are returned via the Standard error (stderr).

  After importing **sys Module** we can use these Standard Streams in the same way you use other files.

Key Word

sys Module: The sys module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment.

```
import sys
sys.stdin.read() – lets program read from keyboard
sys.stdout.write() – lets program write to the standard output Device
sys.stderr.write() – lets you write the error
```

When we start python these files are opened in:

stdin(read mode)

stout(write mode)

stderr(write mode)

Python stderr is similar to stdout because it also directly prints to the console but the main difference is that it only prints error messages.

Relative and Absolute Paths

The absolute path is the full path to some place on your computer. It describes how to access a given file or directory starting from the root of the file system.

The relative path is the path to some file with respect to your current working directory (PWD).

For example, for the file text1.txt which is store in C directory ->folder -> subfolder1 -> subfolder2.

Absolute path: C:/folder/ subfolder1 / subfolder2/text1.txt

If PWD is C:/folder/ subfolder1 /, then the relative path to text1.txt would be: subfolder2/text1.txt

- **Basic Operations on a Binary File**

Open and Closing a binary file: Binary files store data in the binary format (0's and 1's) which is understandable by the machine. So when we open the binary file in our machine, it decodes the data and displays in a human-readable format.

Opening and closing of binary file is same as text file opening and closing. While opening any binary file we have to specify 'b' in file opening mode. Hence the "rb" mode opens the file in binary format for reading, while the "wb" mode opens the file in binary format for writing.

- **Pickle Module**

It is very difficult for us to write several types of objects into the binary file. This is very difficult task, particularly if some of the objects can have variable lengths. We cannot read the contents of the file later.

Key Concepts

- When a file is opened in write mode then cursor at starting index. That's why if we start writing file in write mode then it starts to write file from beginning and over write the existing data in file.

But in append mode ie. "a" mode, the cursor is always placed at the last index.

- PWD + relative path = absolute path.
- Pickle library is developed using the C programming language like the python interpreter is.

It can save complex Python data structures.

To overcome this problem we use pickle module. Pickle module provides us with the ability to **serialize** and **de-serialize** objects, i.e., to convert objects into bitstreams which can be stored into files and later be used to reconstruct the original objects.

Serialization and De-serialization process is also called **pickling** and **un-pickling** respectively.

Key Words

Serialization: Serialization is the process of transforming data or an object in memory (RAM) to a stream of bytes called byte streams. These byte streams in a binary file can then be stored in a disk or in a database or sent through a network.

De-serialization: De-serialization or unpickling is the inverse of pickling process where a byte stream is converted back to Python object.

dump and load Methods:

Pickle module contains a dump() function to perform pickling and pickle module contains load() function to perform unpickling.

Syntax of dump():

```
dump(data_object, file_object)
```

Where data_object is the object that has to be dumped to the file with the file handle named file_ object.

Syntax of load():

```
store_object = load(file_object)
```

Here, the pickled Python object is loaded from the file having a file handle named file_object and is stored in a new file handle called store_object.

Read from and Write/Create into a binary file: Use the 'rb' mode in the open() function to read a binary files.

Python provided read ( ) and write( ) methods works with string parameters and will not directly work with binary files. Conversion of data at the time of reading and writing is required.

dump() method is used to write the objects in a binary file and load() method is used to read data from a binary file.

Search: There is no any pre-defined function available in python for searching records in binary file in python.

Searching in binary file involves reading of the binary file and then comparing each record with our given value, thus we are bound to use the linear search method.

For example, consider the following program of searching the name from a file.

```
import pickle
name = input(Write the name:')
file = open("employee.dat", "rb")
list = pickle.load(file)
```

```
file.close()
get = 0
for i in list:
    if name in i['name']:
            get = 1
if get==1
    print("Found in binary file" )
else
    print("Not found")
```

► **Append and Update operations in a binary file:**

To append the data to the binary file, open a file in 'ab' mode. A file opened in append mode will retain the previous records and append the new records at the end of the file.

In order to update a binary file, one must know the position of file pointer. To Check out the position of the file pointer, we use two file pointer location functions: ***tell()*** and ***seek()***.

► **Accessing and Manipulating Location of File Pointer**

Python provides two functions to manipulate the position of file pointer and thus user can read and write from desired position.

- **tell( ) function:** The tell( ) function returns the current position of the file pointer in the file.

  Syntax:

  <fileobject>.tell( )
- **seek( ) function:** The seek ( ) function changes the position of the file pointer by placing the file pointer at the specified position in the open file.

  Syntax:

  <fileobject>.seek(offset[,mode])

  Where

  offset =======>is number specifying number of bytes

  mode =======> is number 0 or 1 or 2

  0 for beginning of file

  1 current position of file pointer

  2 end of file

► **Import CSV Module**

In Python, to work with CSV files, we need to import the CSV module, an inbuilt module. To do this, we need to use the "import" keyword before "csv" to support CSV files in python.

The CSV module includes all the necessary functions built in. They are:

- csv.reader
- csv.writer
- csv.register_dialect
- csv.unregister_dialect
- csv.get_dialect
- csv.list_dialects
- csv.field_size_limit

► **Basic operations on a CSV file**

- **Open a CSV file:** The CSV file is opened as a text file with Python's built-in open() function, which returns a file object.
- **Close a csv file:** Closing of binary file is same as text file closing.
- **Read from a csv file:** Reading from a CSV file is done using the reader object. The CSV file is opened as a text file with Python's built in open() function, which returns a file object.

  Example:

```
import CSV
with open("Employee.txt") as CSV_file:
CSV_reader=CSV.reader(CSV_file, delimiter =',')
   line_conut = 0
```

```
for row in CSV_reader:
            if line_count == 0:
            Print(f' column names are{",".join(row)}) line_count + = 1
        else:
            print(f'\t{row[0]} works in the {row[1]} department, and was born in {row[2]}.') line_
            count + = 1
print(f'Processed {line_count} lines.)
```

- **Write into and read from a csv file using csv.reader ( ) and csv.writerow( )**

  For reading, we open the file in 'r' mode and create newFile like object,further we create newfilereader object using csv.reader()method to read each row of the file.

  The **csv.reader** method returns a reader object which iterates over lines in the given CSV file.

- **Optional Python CSV reader Parameters**

  The reader object can handle different styles of CSV files by specifying additional parameters, some of which are shown below:

  - Delimiter specifies the character used to separate each field. The default is the comma (',').
  - quotechar specifies the character used to surround fields that contain the delimiter character.
  - The default is a double quote (' " ').
  - escapechar specifies the character used to escape the delimiter character, in case quotes are not used. The default is no escape character.

  Similarly, for writing, we open file in 'w' writing mode using open() method which create newFile like object.

  When you have a set of data that you would like to store in a CSV file you have to use **csv.writer()** function. To iterate the data over the rows(lines), you have to use the **csv.writerow()** function.

  e.g

```
import CSV
with open ('Employee_file.CSV', mode='w') as Employee_file:
Employee_writer = CSV.writer (Employee_file, delimiter = ',' quotechar = ' " ' quoting = CSV. Quote_Minimal)
Employee_writer.writerow(['Rahul', 'Manager', 'April'])
Employee_writer.writerow(['Neha', 'IT', 'June'])
```

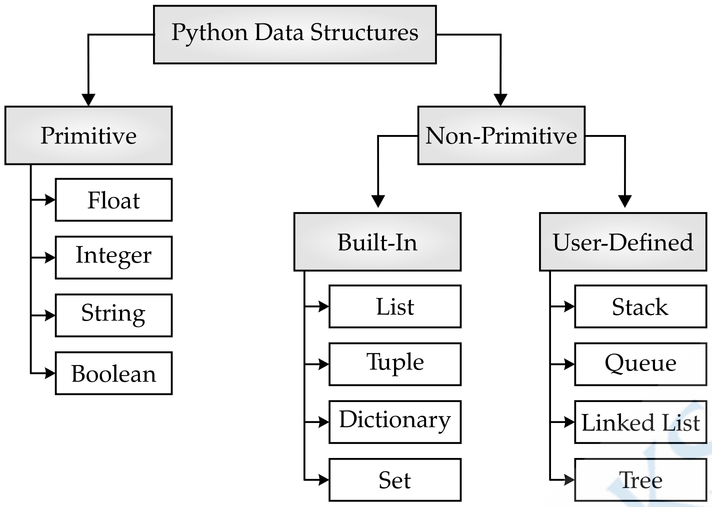
Key Words

- CSV stands for Comma Separated Values.
- Pickle module can be used to store any kind of object in file as it allows us to store Python objects with their structure.
- File Handle serve as a link to a file residing on the computer.
- File Mode governs the type of operations possible in the operand file. The default mode is read ('r')
- flush () function forces the writing of data on disc still pending in output buffers.

CHAPTER-4
DATA STRUCTURE

Revision Notes

- **Data Structure** is a named group of data of different data types which is stored in a specific way and can be processed as a single unit. A Data Structure has well-defined operations, behaviour and properties.
- Python has implicit support for Data Structures which enable you to store and access data. These structures are called List, Dictionary, Tuple and Set.
- Python allows its users to create their own Data Structures enabling them to have full control over their functionality. The most prominent Data Structures are Stack, Queue, Tree, Linked List.



- **List (Linear Data Structure)**
  It is a collection of items and each item has its own index value. Index of first item is 0 and the last item is n-1. Here n is number of items in a list.
- There is also **negative indexing** which starts from -1 enabling you to access elements from the last to first.
- **Creating a list:** To create a list, you use the square brackets and add elements into it accordingly. The 2nd method is to call the Python list built-in function by passing the items to it.
- **Access Items from a List:** To access values in lists, use the square brackets for slicing along with the index or indices to obtain value available at that index.
- **Working with List or List Operations**
  - **Concatenation:** It concatenates the list mentioned on either side of the operator (+).
  - **Repetition:** The repetition operator(*) enables the list elements to be repeated multiple times.
  - **Membership:** It returns true if a particular item exists in a particular list otherwise false.
  - **Updating List values and Adding elements to the list:** Since Lists are mutable, and their values can be updated by using the slice and assignment operator.
    Python also provides append() and insert() methods, which can be used to add values to the list.
  - **Removing elements from the list:** Python provides the remove() function which is used to remove the element from the list. The list elements can also be deleted by using the del keyword.
  - **Traversing List**
    Traversing a list is a process to iterate the particular element of a list. With the help of traversing, one can iterate all or particular element of list. A list can be iterated by using looping statement.
  - **Python List Built-in functions**

Some useful functions of lists:

Function	Description
list.append()	Add an Item at end of a list
list.extend()	Add multiple Items at end of a list
list.insert()	Insert and Item at a defined index
list.remove()	remove an Item from a list
del list[Index]	Delete an Item from a list
list.clear()	empty all the list
list.pop()	Remove an Item at a defined index
list.index()	Return index of first matched item
list.sort()	Sort the items of a list in ascending or descending order
list.reverse()	Reverse the items of a list
len(list)	Return total length of the list.
max(list)	Return item with maximum value in the list.
min(list)	Return item with min value in the list.
list(seq)	Converts a tuple, string, set, dictionary into list.

► **Stack**

► A Stack is a linear Data Structure implemented in LIFO (Last In First Out) manner.

► Insertions and deletions both operations occur only at only one end known as TOP.

► We can create a Stack in python using the built-in List Data Structure which comes with methods to simulate Stack

► To implement a Stack, we need two simple operations:

- push - It adds an element to the top of the Stack.
- pop - It removes an element from the top of the Stack.

Applications of Stack

- Expression Evaluation
- Expression Conversion
- Parenthesis Checking
- String Reversal
- Function Call

► **Operations on Stack**

The Stack supports the following standard operations:

- **push:** Pushes an item at the top of the Stack. It returns an error if the Stack is **overflow**.
- **pop:** Remove and return the item from the top of the Stack. It produces an error if the Stack is appeared to be **underflow**.
- **peek:** Returns the item at the top of the Stack without removing it.
- **size:** Returns the total number of items in the Stack.
- **isEmpty:** Checks whether the Stack is empty.
- **isFull:** Checks whether the Stack is full.

Key Words

Overflow: Overflow refers to condition when one tries to PUSH an item in Stack which is full already.

Underflow: Underflow refers to condition when one tries to POP an item from an empty Stack.

► **Implementation of Stack**

The implementation of Stack using list in Python program is the easiest of all programming languages.

It offers a convenient set of methods to operate lists as Stack.

Python's built-in Data Structure list can be used as a Stack. Instead of push(), append() is used to add elements to the top of the Stack while pop() removes the element in LIFO order.

Key Words

► Postfix notation refers when operator is placed after its operands e.g. ab+.

► Prefix notation refers when operator is placed before its operands e.g. +ab.

► Infix notation refers when operator is placed between its operands e.g. a+b.

► While conversion of an Infix notation to its equivalent Prefix/Postfix notation, only operators are Pushed onto the Stack.

► **Nested lists:** When one or more elements of list is another list, it is called a nested list.

UNIT – II: COMPUTER NETWORKS

CHAPTER-5

DATA COMMUNICATION & NETWORK TOPOLOGIES

Topic-1

Networking and Communication Terminologies

Concepts Covered
- *Basics of Networking*
- *Circuit and Packet Switching Techniques*
- *Data Communication Basics*
- *Various Transmission Media*

Revision Notes

► **Computer Network**

A computer network is a group of two or more interconnected computer systems. You can establish a network connection using either cable or wireless media.

Every network involves hardware and software that connects computers and tools.

► Evolution of Networking

(A) **ARPANET:** In 1969, US Government formed an agency named ARPANET (Advanced Research Projects Agency Network) to connect computers at various universities and defense agencies. The main objective of ARPANET was to develop a network that could continue to function efficiently even in the event of a nuclear attack.

(B) **NSFNET:** The term "NSFNET" refers to a program of coordinated, evolving projects sponsored by the National Science Foundation that was initiated in 1985 to support and promote advanced networking among U.S. research and education institutions.

(C) **INTERNET (INTER-connection NETwork):** The Internet is a worldwide network of computer networks.

(D) **Interspace:** It is a software that allows multiple users in a client server environment to communicate with each other to send and receive data of various types such as data files, videophone, audio and textual data. Interspace gives the most exceptional type form of communication available on the Internet today.

► **Switching Techniques:** These are used for transmitting data across networks.

(A) **Circuit switching:** In the circuit switching technique, first the complete end to end transmission path between the source and the destination computers is established and then the message is transmitted through the path. The main advantage of this technique is guaranteed delivery of the message which is mainly used for voice communication.

(B) **Message Switching:** In the message switching technique, no physical path is established between sender and receiver in advance. This technique follows the store and forward mechanism.

(C) **Packet Switching:** In this technique this message is broken into small data packets. In this switching technique, fixed size of packet can be transmitted across the network.

Key Word

Switching: Switching is the technique by which nodes control or switch data to transmit it between specific points on a network.

► **Comparison between the various switching techniques:**

Criteria	Circuit switching	Message Switching	Packet Switching
Path established in advance	Yes	No	No
Store and forward technique	No	Yes	Yes
Message follows multiple routes	No	Yes	Yes

► **Concept of Communication**

Data can be any text, image, audio, video, and multimedia files. Communication is an act of sending or receiving data.

Thus, data communication refers to the exchange of data between two or more networks or connected devices. These devices must be capable of sending and receiving data over a communication medium.

Examples of such devices include personal computers, mobile phones, laptops, etc.

Components of Data Communication

Sender, receiver, communication medium, the message to be communicated, and certain rules called protocols to be followed during communication. The communication media is also called transmission media.

The role of these five components in data communication is as follows:

(A) Sender: A sender is a computer or any such device which is capable of sending data over a network.

(B) Receiver: A receiver is a computer or any such device which is capable of receiving data from the network. In computer communication, the sender and receiver are known as nodes in a network.

(C) Message: It is the data or information that needs to be exchanged between the sender and the receiver.

(D) Communication media: It is the path through which the message travels between source and destination. It is also called medium or link which is either wired or wireless.

(E) Protocols: It is a set of rules that need to be followed by the communicating parties in order to have successful and reliable data communication.

Data communication Terminologies

(A) Concept of Channel : A data channel is the medium used to carry information or data from one point to another.

(B) Band: It is the unit to measure the data transmission speed. It is equivalent to bps (bits per second).

(C) Band-width: The maximum volume of data that can be transferred over any communication channel at a given point of time is known as the bandwidth. In analog systems, it is measured in hertz (Hz) and in digital systems, it is measured in bits per second (Bps).

Hertz are commonly expressed in multiples: kilohertz (103 Hz, kHz), megahertz (106 Hz, MHz), gigahertz (109 Hz, GHz), terahertz (1012 Hz, THz).

(D) Data Transfer Rate: The amount of data transferred per second by the communication channel from one point to another is known as the data transfer rate. It is measured in bits per second (bps), bytes per second (Bps).

1 kbps = 2^{10} Bps	1 GBps =2^{30} Bps
1 MBps = 2^{20} Bps	1 TBps = 2^{40} Bps

Transmission Media

Transmission media of a network refers to the connecting media used in the network. It can be broadly defined as anything that can carry information from a source to destination.

(A) Twisted Pair Cable : It consists of two identical 1mm thick copper wires insulated and twisted together. The twisted pair cables are twisted in order to reduce crosstalk and electromagnetic induction.

Advantages:

- It is easy to install and maintain.
- It is inexpensive.

Disadvantages:

- It is incapable to carry a signal over long distance without the use of repeaters.
- Due to low bandwidth, these are unsuitable for broadband applications.

(B) Coaxial Cable : It consists of a solid wire core surrounded by one or more coil or braided wire shields, each separated from the other by some kind of plastic insulator. It is mostly used in the cable wires.

Advantages:

- Data transmission rate is better than twisted pair cables.
- It provides a cheap means of transporting multi-channel television signals around metropolitan areas.

Disadvantages:

- Expensive than twisted pair cables.
- Difficult to manage and reconfigure.

(C) Optical fiber : It consists of thin glass fibres that can carry information in the form of visible light.

Advantages:

- Transmit data over long distance with high security.
- Data transmission speed is high.
- Provide better noise immunity.
- Bandwidth is up to 10 Gbps.

Disadvantages:

- Expensive as compared to other guided media.
- Need special care while installation.

(D) Infrared: The infrared light transmits data through the air and can propagate within a room, but will not penetrate walls. It is a secure medium of signal transmission. The infrared transmission has become common in TV remote, automotive lift doors, wireless speakers, etc.

Advantages:

- Power consumption is used.
- Circuitry cost is less.
- Secure mode of transmission.

Disadvantages

- Limited to a short range.
- Can be blocked by common materials like walls, people, plants, etc.

(E) Radio wave : It is an electromagnetic wave with a wavelength between 0.5 cm to 30,000m. The transmission making use of radio frequencies is termed as radio-wave transmission.

Advantages:

- Radio wave transmission offers mobility.
- It is cheaper than laying cables and fibers.
- It offers ease of communication over difficult terrain.

Disadvantages:

- Radio wave communication is insecure communication.
- Radio wave propagation is susceptible to weather effects like rains, thunder storms etc.

(F) Microwave : The microwave transmission is a line of sight transmission. Microwave signals travel at a higher frequency than radio waves and are popularly used for transmitting data over long distances.

Advantages:

- It is cheaper than laying cable or fiber.
- It has the ability to communicate over oceans.

Disadvantages:

- Microwave communication is an insecure communication.
- Signals from antenna may split up and get transmitted in different way to different antenna which leads to reduce in signal strength.
- Microwave propagation is susceptible to weather effects like rains, thunder storms, etc.
- Bandwidth allocation is extremely limited in case of microwaves.

Key Words

- **Backbone:** Referring to the Internet, a central network that provides a pathway for other networks to communicate.
- **Broadband:** A transmission system capable of carrying many channels of communication simultaneously by modulating them on one of several carrier frequencies.
- **Congestion:** Traffic in excess of network capacity.
- **Data Link:** The physical connection between two devices such as Ethernet, Local Talk or Token Ring that is capable of carrying information in the service or networking protocols such as AppleTalk, TCP/IP or XNS.
- **Destination Address:** Address of network device that is receiving data.
- **Gbps:** Gigabits per second
- **GHz:** Giga Hertz
- **KBPS:** A unit of measure used to describe the rate of data transmission equal to 1000 bits per second.
- **MBPS:** A unit of measure used to describe the rate of data transmission equal to one million bits per second.
- **Packet:** A discrete chunk of communication in a per defined format.

Topic-2 Network Devices, Topologies and Protocols

Concepts Covered

- *Different Network Devices,*
- *Various types of Network Topologies,*
- *Advantages and Disadvantages of network Topologies,*
- *Types of Network(PAN, LAN, MAN, WAN),*
- *Types of protocol.*

Revision Notes

Network Devices

These are units that mediate data in a computer network and are also called network equipment. Some of them are following:

- **Modem:** A MODEM (Modulator Demodulator) is an electronic device that enables a computer to transmit data over telephone lines. There are two types of modems namely, internal modem and external modem.
- **RJ45 Connector:** The RJ-45 (Registered Jack) connectors are the plug-in devices used in the networking and telecommunication applications. They are used primarily for connecting LANs, particularly Ethernet.
- **Ethernet card:** It is a hardware device that helps in connection of nodes within a network.
- **Hub:** It is a hardware device used to connect several computers together. Hubs can be either active or passive. Hubs usually can support 8,12 or 24 RJ45 ports.
- **Switch:** A switch (switching hub) is a network device which is used to interconnect computers or devices in a network. It filters and forwards data packets across a network. The main difference between a hub and a switch is that hub replicates what is received on one port onto all the other ports while switch keeps a record of the MAC addresses of the devices attached to it.
- **Router:** It is a hardware device which is designed to take incoming packets, analyse packets, moving and converting packets to the another network interface, dropping the packets, directing packets to the appropriate locations, etc.
- **Gateway:** It is a device that connects dissimilar networks.
- **Repeater:** It is a network device that amplifies and restores signals for long distance transmission.
- **Wi-Fi card:** Connects to your laptop either in your USB port or wider card slot. This card generally is geared to a particular Wi-Fi network, so to use it you must be in range of a wireless internet signal dedicated to that network.

Network Topologies

Topology refers to the way in which the workstations attached to the network are interconnected.

Some common network topologies are as follows:

Bus Topology: It uses a common single cable to connect all the workstations. Each computer performs its task of sending messages without the help of the central server. However, only one workstation can transmit a message at a particular time in the bus topology.

Advantages:

- Easy to connect and install.
- Involves a low cost and installation time.
- Can be easily extended.

Disadvantages:

- The entire network shuts down if there is a failure in the central cable.
- Only a single message can travel at a particular time.
- Difficult to troubleshoot an error.

Star Topology: It is based on a central node which acts as a hub. A star topology is common in-home networks where all the computers connect to the single central computer, using a hub.

Advantages:

- Easy to troubleshoot.
- A single node failure does not affect the entire network.
- Fault detection and removal of faulty parts is easier.
- In case a workstation fails, the network is not affected.

Disadvantages:

- Difficult to expand.
- Longer cable is required.
- The cost of the hub and the longer cables make it expensive over others.
- In case hub fails, the entire network fails.

Tree Topology: It combines the characteristics of the linear bus and star topologies. It consists of groups of star configured workstation connected to a bus backbone cable.

Advantages:

- Eliminates network congestion.
- The network can be easily extended.
- Faulty nodes can be easily isolated from the rest of the network.

Disadvantages:

- Uses large cable length.
- Requires a large amount of hardware components and hence is expensive.
- Installation and reconfiguration is very difficult.

Network Types

- **LAN(Local Area Network):** It is a network that is confined to a relatively small area. It is generally limited to a geographic area such as writing lab, school or building. It is generally privately owned network over a distance not more than 5 km.
- **MAN (Metropolitan Area Network):** It is a network that covers a group of nearby corporate offices in a city and might be either private or public.
- **WAN (Wide Area Network):** These are the networks spread over large distances, say across countries or even continents through cabling or satellite uplinks.
- **PAN(Personal Area Network):** It is a computer network organized around an individual person. It generally covers a range of less than 10 meters. PAN can be constructed with cables or wirelessly.

Network Protocol

A protocol means the rules that are applicable for a network. It defines the standardized format for data packets, techniques for detecting and correcting errors and so on. A protocol is a formal description of message formats and the rules that two or more machines must follow to exchange those messages.

Types of protocols are:

- **Hypertext Transfer Protocol (HTTP):** It is a communication protocol for the transfer of information on the Internet and world wide web. HTTP is a request/response standard between a client and a server. A client is the end-user while, the server is the website.
- **FTP (File Transfer Protocol):** It is the simplest and most secure way to exchange files over the Internet. The objectives of FTP are:
  1. To promote sharing of Files (computer programs and/or data).
  2. To encourage indirect or implicit use of remote computers.
  3. To shield a user from variations in file storage systems among different hosts.
  4. To transfer data reliably and efficiently.
- **TCP/IP (Transmission Control Protocol/ Internet Protocol):** TCP is responsible for verifying the correct delivery of data from client to server. Data can't be lost in the intermediate network. TCP adds support to detect errors or lost data and to trigger retransmission until the data is correctly and completely received.

  IP is responsible for moving packet of data from node to node. IP forwards each packet based on a four-byte destination address (the IP number). The Internet authorities assign ranges of numbers to different organisations. The organisations assign groups of their numbers to departments. IP operates on gateway machines that moves data from department to organization, then to region and then around the world.
- **SMTP(Simple Mail Transfer Protocol):** It is a standard protocol for email services on TCP/IP network that provides ability to send and receive e-mail.
- **POP3 (Post Office Protocol Version3):** It is a message access protocol which allows client to fetch an e-mail from remote mail server.
- **Telnet** is a network protocol used to virtually access a computer and to provide a two-way, collaborative and text based communication channel between two machines.
- **PPP (Point - to - Point Protocol):** It is a communication protocol of the data link layer that is used to transmit multiprotocol data between two directly connected (point-to-point) computers. It is a byte - oriented protocol that is widely used in broadband communications having heavy loads and high speeds.
- **HTTPS (Hypertext transfer protocol secure):** It is the secure version of HTTP, which is the primary protocol used to send data between a web browser and a website. HTTPS is encrypted in order to increase security of data transfer.
- **VoIP (Voice over Internet Protocol):** It is a proven technology that lets anyone place phone calls over an internet connection. With the rise of broadband, VoIP has become the definitive choice of phone service for consumers and businesses alike.

Key Words

- **Bridge:** A data link layer device that limits traffic between two network segments by filtering the data between them based on hardware addresses.
- **Token Ring:** A ring topology network that uses token passing for MAC.
- **Transport Protocol:** The protocol layer of the OSI-7 layer model that is concerned with management of data flow between source and destination.

Topic-3 Web Services

Concepts Covered • *Different types of services used in network.*

Revision Notes

- **Web Services**
  - **WWW:** The world wide web or W3 or simply the web is a collection of linked documents or pages, stored on millions of computers and distributed across the Internet.

Key Word

Web Service: A Web service is a software service used to communicate between two devices on a network.

- **HTML (HyperText Markup Language):** It is a computer language that describes the structure and behavior of a web page. This language is used to create web pages.
- **XML (Extensible Markup Language):** It is a meta language that helps to describe the markup language.
- **Domain Name:** It is a unique name that identifies a particular website and represents the name of the server where the web pages reside.
- **URL(Uniform Resource Locator):** It is a means to locate resources such as web pages on the Internet. URL is also a method to address the web pages on the Internet. There are two types of URL namely, Absolute URL and Relative URL.
- **Website:** A collection of related web pages stored on a web server is known as a website.
- **Web browser:** A software application that enables to browse, search and collect information from the web is known as web browser.
- **Web server:** The web pages on the Internet are stored on the computers that are connected to the Internet. These computers are known as web servers.
- **Web hosting:** Web hosting or website hosting is the service to host, store and maintain websites on the world wide web.

CHAPTER-6

DATABASE CONCEPTS

Revision Notes

Database Concepts

- **Database Concepts and Needs:**
  - A database is tabular organization of data. Each table comprises of rows (records) and columns (attributes). Each record contains values for the corresponding attributes. The values of the attributes for a record are interrelated. For example, different cases have different values for the same specifications (length, color, engine capacity, etc).
  - The database oriented approach supports multiple views of the same data. For example, a clerk may only be able to see his details, whereas the manager can view the details of all the clerks working under him.
  - In database oriented approach, we store the common data in one table and access it from the required tables. Thus, the redundancy of data decreases.

- Multiple views of the same database may exist for different users. This is defined in the view level of **abstraction**.

Key Word

Abstraction: Data Abstraction in dbms refers to the process of hiding irrelevant details from the user.

There are mainly three levels of data abstraction: Internal Level, Conceptual or Logical Level or External or View level ·

- The logical level of abstraction defines the type of data that is stored in the database and the relationship between them.
- The design of the database is know as the database schema.
- The instance of the database is the data contained by it at that particular moment.
- The database administrator has the total control of the database and is responsible for setting up and updating the database.

Need for a Database

Data is stored in the form of files. A number of application programs are written by programmers to insert, delete, modify and retrieve data from these files. New application programs will be added to the system as the need arises.

- A data model is the methodology used by a particular DBMS to organize and access the data.
- Hierarchical, Network and Relational model are the three popular data models. However, the relational model is more widely used.

Hierarchical Model

- The hierarchical model was developed by IBM in 1968.
- The data is organized in a tree structure where the nodes represent the records and the branches of the tree represent the fields.
- Since the data is organized in a tree structure, the parent node has the links to its child nodes. This is called (PCR) parent child relation.
- If we want to search a record, we have to traverse the tree from the root through all its parent node to reach the specific record. Thus, searching for a record is very time consuming.
- The hashing function is used to locate the root.
- SYSTEM 2000 is an example of hierarchical database.

Network Model

- Data is represented by collection of records and relationships among data are represented by links.
- Recording of relationship in the network model is implemented by using pointers.
- Recording of relationship implementation is very complex since pointers are used. It supports many-to-many relationship and simplified searching of record, since a record has many access paths.
- DBTG CODASYL was the first network database.

Relational Model

- The Relational model, organizes data in the form of independent tables (consisting of rows and columns) that are related to each other.
- A relation is a two-dimensional table. It contains number of rows(tuples) and columns(attributes).
- A table consists of a number of rows (records/tuples) and columns (attributes). Each record contains values for the attributes.
- A single entry in a table is called a Tuple or Record or Row.
- A database attribute is a column or field in a database table.
- The **degree** of the table denotes the number of columns.
- **Cardinality** is defined as the number of rows in a table.
- In a database, a domain is a column that contains a data type. A domain in the relational model is said to be atomic, if it consists of indivisible units. For example, name is not atomic since it can be divided into first name and last name.
- E.F. Codd laid down 12 rules (known as Codd's 12 rules), that outline the minimum functionality of a RDBMS. A RDBMS must comply with at least 6 rules.
- A super key is a type of attribute that collectively identifies an entity in an entire set. For example, the bank account number is a super key in the bank accounts table.

- A candidate key (also known as primary key) is the smallest subset of the super key for which there does not exist a proper subset that is a super key.
- Out of the multiple candidate keys, only one is selected to be the primary key and the remaining are alternate keys.
- A foreign key is the primary key of a table that is placed into a related table to represent one-to-many relationship among these tables.
- A primary key is a set of one or more attributes that can uniquely identify the relation.
- Primary key uniquely identifies the records in the table.

Key Words

- **DBMS:** It stands for Database Management System that enables users to define, create and maintain the database and provides controlled access to this database.
- **DBA:** DBA is Database Administrator that has the central control over the system.
- **RDBMS:** RDBMS stands for Relational Database Management System.
- **Meta Data:** It means data about data i.e., a logical description of the structure of a data.
- **File Processing System:** It is a collection of new data files stored in the hard drive of a system.

CHAPTER-7

STRUCTURED QUERY LANGUAGE (SQL)

Revision Notes

Structured Query Language

1. **Structured Query Language (SQL)**
   - When a user wants to get some information from a database file, he can issue a query.
   - A **query** is a user-request to retrieve data or information with a certain condition.
   - SQL is a query language that allows user to specify the conditions, (instead of algorithms).

Key Word

QUERY: A query is a request for data or information from a database table or combination of tables.

Advantages of Using SQL

- High speed
- No coding
- Well defined standards
- Portability
- Interactive language
- Multiple data view

2. **Types of SQL commands**
   - **Data Definition Language (DDL) Commands:** All the commands used to create, modify or delete physical structure of an object like table. **e.g.,** Create, Alter, Drop.
   - **Data Manipulation Language (DML) Commands:** All the commands used to modify contents of a table comes under this category. **e.g.,** Insert, Delete, Update
   - **Transaction Control Language (TCL) Commands:** These commands are used to control transaction of DML commands. **e.g.,** Commit, Rollback.

3. **Data type of Attribute**

   Data type indicates the type of data value that an attribute can have.

   - **CHAR(n):** Specifies character type data of length n where n could be any value from 0 to 255.

- **VARCHAR(n):** Specifies character type data of length 'n' where n could be any value from 0 to 65535. But unlike CHAR, VARCHAR is a variable-length data type.
- **INT:** INT specifies an integer value. Each INT value occupies 4 bytes of storage.
- **FLOAT:** Holds numbers with decimal points. Each FLOAT value occupies 4 bytes.
- **DATE:** The DATE type is used for dates in 'YYYY-MM-DD' format. YYYY is the 4 digit year, MM is the 2 digit month and DD is the 2 digit date.

4. Basic structure of an SQL query

- **General structure** SELECT, ALL/DISTINCT, *, AS, FROM, WHERE
- **Comparison** IN, BETWEEN, LIKE "% _"
- **Grouping** GROUP BY, HAVING, COUNT(), SUM(), AVG(), MAX(), MIN()
- **Display order** ORDER BY, ASC/DESC
- **Logical operators** AND, OR, NOT
- **Create Database:** To create a database, use the CREATE DATABASE statement.
  Syntax–CREATE DATABASE database_name;
- **Show Database:** To display the name of all databases, use SHOW command.
  Syntax– SHOW DATABASES;
- **USE database:** To use the database follow the syntax
  USE Database_name;
- **Drop database**
  To drop database, DROP DATABASE statement is used.
  Syntax DROP DATABASE database__name;
- **Show tables** To show all tables in current database, SHOW TABLES command is used.
  Syntax SHOW TABLES;

5. Constraint

- Constraint is a condition applicable on a field or group of fields.
- Two types of constraint:
  - **Column constraint:** Apply only to individual column.
  - **Table constraint:** Apply to group of columns.
  - **Different constraints:**
    Unique Constraint-Primary Key constraint.
    Default Constraint-Check constraint.

Primary and Foreign Key

Primary key: A table can have only one primary key. The primary key column has a unique value and doesn't store repeating values. A Primary key can never take NULL values.

Syntax of creating the table with the Primary key specified.

```
CREATE TABLE tableName (
   col1 data type NOT NULL,
   col2 data type NOT NULL,
    ................
   PRIMARY KEY (col1)
);
```

Foreign key: A foreign key is a field or collection of fields in a table that refers to the Primary key of the other table.

Syntax of creating a table with a foreign key.

```
CREATE TABLE childTable (
   col1 Datatype NOT NULL,
   col2 Datatype NOT NULL,
   col3 Datatype ,
   ...........
   PRIMARY KEY (col1),
  FOREIGN KEY (col3) REFERENCES parentTable (parent_Primary_key)
);
```

Applying Constraint

Example: Create a student table with student id, student name, father's name, age, class, address.

```
CREATE TABLE student
sid char (4) PRIMARY KEY,
sname char (20) NOT NULL,
fname char (20),
age number (2) CHECK (age<20),
class char (5) NOT NULL,
address char (50));
```

6. **SELECT Command**

The SELECT command is a query that is given to produce certain specified information from the database table.

Syntax:

```
SELECT <column-name>,[<column-name>,......]
FROM <table-name>;
```

Example: Write a query to display the name and salary of the employee in emp table.

```
SELECT ename, sal
FROM emp;
```

Variations of SELECT command:

(i) Selecting specific Rows ....WHERE clause

Syntax:

```
SELECT <column-name> [<column-name>.......]
FROM <table-name>
WHERE <condition>;
```

Example: Display the codes, names and salary of employees who belong to 'Manager' category.

```
SELECT ecode, ename, sal
FROM emp
WHERE job="MANAGER";
```

(ii) Searching for NULL (IS NULL command)

The Null value in a column can be searched for in a table using IS NULL in the WHERE Clause

Syntax:

```
SELECT......<column-name>, <column-name>......
FROM <table-name>
WHERE <column-name> IS NULL;
```

Example: Display the codes, names and jobs of employees whose DeptNo is Null.

```
SELECT ecode,ename, job
FROM emp
WHERE DeptNo IS NULL;
```

(iii) IS NOT NULL Command

Example: Display the names and jobs of those employees whose DeptNo is not Null.

```
SELECT ename, job FROM emp
WHERE DeptNo IS NOT NULL;
```

(iv) Sorting Result-ORDER BY Clause

The resulting column can be sorted in ascending and descending order using the ORDER BY clause.

Syntax:

```
SELECT <column-name>, <column-name>.......
FROM <table-name>
WHERE <condition>
ORDER BY <column-name>ASC/DESC;
```

Example:

Display the list of employees in descending order of employee code, who is manager.

```
SELECT * FROM emp
WHERE job="MANAGER"
ORDER BY ecode DESC;
```

(v) Conditions based on a range

SQL provides a BETWEEN operator that defines a range of values that the column value must fall for the condition to become true.

Example:

SELECT Roll_no, name From

student WHERE Roll_no BETWEEN 100 AND 103;

The given command displays Roll_no and name of those students whose Roll_no lies in the range 100 to 103 (both 100 and 103 are included in the range).

(vi) Conditions based on a list

To specify a list of values, IN operator is used. This operator selects values that match any value in the given list.

Example:

SELECT * FROM student WHERE city IN ('Delhi', 'Agra', 'Gwalior');

The above command displays all those records whose city is any of Delhi, Agra, Gwalior.

(vii) Conditions based on Pattern

SQL provides two wild card characters that are used while comparing the strings with LIKE operator.

(A) percent (%) matches any string.

(B) Underscore(_) matches any one character.

Example:

SELECT Roll_no, name, city FROM student WHERE Roll_no LIKE "%3";

This query displays those records where last digit of Roll_no is 3 and may have any number of characters in front.

Example:

SELECT Roll_no, name, city FROM student WHERE Roll_no LIKE "1_3";

This query displays those records whose Roll_no starts with 1 and second letter may be any letter but ends with digit 3.

(viii) Having clause

The SQL HAVING clause is used in combination with the GROUP BY clause to restrict the groups of returned rows to only those whose the condition is TRUE.

Syntax:

```
SELECT expression1, expression2, ... expression_n,
    aggregate_function (aggregate_expression)
FROM tables
[WHERE conditions]
GROUP BY expression1, expression2, ... expression_n
HAVING condition;
```

(ix) Distinct Clause

The SQL DISTINCT clause is used to remove duplicates from the result set of a SELECT statement.

Syntax:

```
SELECT DISTINCT expressions
FROM tables
[WHERE conditions];
```

7. The INSERT INTO command

The tuples are added to relation using INSERT command of SQL.

Syntax:

```
INSERT INTO <table-name>[<column list>]
VALUES (<value1>,<value2>,<value3>,.....);
```

Example: Enter a new record in student table.

```
INSERT INTO student (sid,sname,fname,age, class,address)
VALUES("101","Mohan","Pawan",15,"8","Jaipur");
```

Output:

```
sid sname fname age   class   address
101 Mohan Pawan 15      8      Jaipur
```

8. **The DELETE FROM command**

   The delete command removes the tuples from the tables. This command removes the entire row from the table and not the individual field. So, no field argument is needed.

   Syntax:

   DELETE FROM <table-name>
   WHERE <condition>;

   Example: Delete all the records of employee whose salary is less than 3000.

   DELETE FROM emp
   WHERE sal<3000;

   To delete all the record from the table.

   Syntax:

   DELETE FROM<table-name>;

9. **The UPDATE command**

   The UPDATE command is used to change some values in existing rows. The UPDATE command specifies the rows to be changed using the WHERE clause, and new data using the SET keyword.

   Example: Update the salary of employee to 5000 whose employee code is 1011.

   UPDATE emp
   SET sal=5000
   WHERE empno=1011;

10. **The ALTER TABLE command**

    The ALTER command is used to change the definition of existing table.

    (i) To add columns to a table

    Syntax

    ALTER TABLE<table-name> ADD <column name> <data type> <size>;

    (ii) To modify existing columns of a table

    Syntax:

    ALTER TABLE <table-name>
    MODIFY (Column-name newdatatype (newsize));

    Example: To modify column job of table emp to have new width of 30 characters.

    ALTER TABLE emp
    MODIFY (job char (30));

11. **The DROP TABLE Command:**

    The DROP command is used to drop the table from the database. For dropping a table, all the tuples should be deleted first *i.e.*, the table should be empty.

    Syntax:

    DROP TABLE <table-name>;

    Example: Drop the student table from the database.

    DROP TABLE student;

    Some Example:

    Ex 1. Write a query on the customers table whose output will exclude all customers with a rating <=100, unless they are located in Shimla.

    SELECT * FROM customers WHERE rating >100 OR city =“Shimla”;

    Ex 2. Write a query that selects all orders except those zeros or NULLs in the amount field.

    SELECT * FROM Orders WHERE amt <>0 AND (amount IS NOT NULL);

    Ex 3. Write a query that lists customers in descending order of rating. Output the rating field first, followed by the customers name and number.

    SELECT rating, cust_name, cust_num FROM customers ORDER BY rating DESC;

    Ex 4. Write a command that puts the following values in their given order, into the salesman table:

    cust_name-Manisha, city-Manali, comm.-NULL, cust_num-1901.

    INSERT INTO salesman (city, cust_name, comm, cust_num) VALUES ("Manali", “Manisha”, NULL, 1901);

    Operators in SQL:

    The following are the commonly used operators in SQL:

(i)	Arithmetic Operators	+, -, *, /, %
(ii)	Relational Operators	=, <, >, <=, >=, <>
(iii)	Logical Operators	Or, And, Not

    - Arithmetic operators are used to perform simple arithmetic operations.

- Relational operators are used when two values are to be compared and logical operators are used to connect search conditions in the WHERE clause in SQL.

Other Operators:

(iv) Range check – between low and high

(v) List check – in

(vi) Pattern check – like, not like (%and _ (under score) are used).

12. SQL Functions:

SQL supports functions which can be used to compute and select numeric, character and date columns of a relation. These functions can be applied on a group of rows. The rows are grouped on a common value of a column in the table. These functions return only one value for a group and therefore, they are called aggregate or group functions.

(i) **SUM ():**It returns the sum of values of a column of numeric type.
e.g., Select sum (salary) from employee;

(ii) **AVG ():** It returns the average of values of a column of numeric type.
e.g., Select avg (salary) from employee;

(iii) **MIN ():** It returns the minimum value of the values of a column or a given relation.
e.g., Select min (salary) from employee;

(iv) **MAX ():** It returns the maximum value of the values of a column or a given relation.
e.g., Select max (salary) from employee;

(v) **COUNT ( ):** It returns the number of rows in a relation.
e.g., Select count (*) from employee;

Join

A join is a query that combines rows from two or more tables. In a join query, more than one tables are listed in FROM clause.

The function of combining data from multiple tables is called joining.

Joins are used when we have to select data from multiple tables is called joining. Joins are used to extract data from two tables, when we need a relationship between certain columns in these tables.

There are different kind of SQL joins:

(i) **Equi-Join**

Equi join is a simple SQL join condition that uses equal sign as a comparison operator.

Syntax

```
SELECT column1, column2, column3
FROM Table1, Table2
WHERE Table1. column1 = Table2. column1;
```

(ii) **Natural Join**

The natural join is a type of equi join and it structured in such a way that, columns with same name of associated tables will appear once only.

Syntax

```
SELECT* FROM Table1
NATURAL JOIN Table2;
```

Key Words

- **Attribute:** A set of properties *e.g.*, name, datatype, size, etc., used to characterise the data items of entities. A group of attributes constructs an entity-type (or table), i.e.: all values of a certain column must confirm to the same attributes. Attributes are optionally complemented by constraints.
- **Column:** A set of values of a single table which resides on the same position within its rows.
- **Constraint:** Similar to attributes constraints define rules at a higher level, data items must confirm to. e.g.: null, primary and foreign key, uniqueness, default value, user-defined-criteria like STATUS < 10.
- **Database:** A set of tables. Those tables contain user data and the data dictionary.
- **Data Control Language (DCL):** A class of statements which defines the access rights to data, e.g: GRANT , REVOKE .
- **Data Definition Language (DDL):** A class of statements which defines logical and physical design of a database, e.g. CREATE, drop.
- **Data Manipulation Language (DML):** A class of statements which retrieves and manipulates data, e.g.: SELECT, INSERT, UPDATE, DELETE, COMMIT, ROLLBACK.

- **Relational Model:** A database in which inter-table relationships are primarily organized through common data columns which define a one-to-many relationship between a row of the primary key table and one or more rows of the matching foreign key table. Equi-joins relate tables that have matching primary/foreign key values, but other comparisons (relationships) may be defined. Besides describing how the database tables are related, the relational model also defines how the related data can be accessed and manipulated. SQL is the most commonly used relational model database language.
- **Relationship:** A reference between two different or the same entity. References are not implemented as links. They base upon the values of the entities.
- **Row:** One record in a table containing information about one single entity. A row has exactly one value for each of its columns - in accordance with First Normal Form. This value may be NULL.
- **Statement:** A single command which is executed by the DBMS. There are 3 main classes of statements: DML, DDL and DCL.
- **Table (=Relation):** A set of rows of a certain entity-type, i.e. all rows of a certain table have the same structure.
- **Transaction:** A logical unit of work consisting of one or more modifications to the database. The ACID criterium must be achieved. A transaction is either saved by the COMMIT statement or completely cancelled by the ROLLBACK statement.
- **Value:** Implementation of a single data item within a certain column of a certain row.
- **View:** A virtual table containing only its definition and no real data. The definition consists of a query to one or more real tables or views. Queries to the view are processed as queries to the underlying real tables.

CHAPTER-8

INTERFACE OF PYTHON WITH SQL DATABASE

Revision Notes

- Python's standard for database interfaces is the Python DB-API.
- Python Database API supports a wide range of database servers such as GadFly, mSQL, MySQL, Oracle, Sybase etc.
- You need to download separate DB-API module for each database you need to access.
- DB-API provides a minimal standard for working with databases.
- MySQLdb is an interface for connecting to a MySQL database servers from Python.
- Connect method of MySQLdb interface is used to create a connection object using MySQLdb module.

Key Word

Cursor: Cursor class is an instance using which you can invoke methods that execute SQLite statements, fetch data from the result sets of the queries. You can create Cursor object using the cursor() method of the Connection object/class.

commit() method sends a COMMIT statement to the MySQL server, committing the current transaction. Since by default Connector/Python does not auto commit, it is important to call this method after every transaction that modifies data for tables that use transactional storage engines.

- A **cursor** is a Python object that enables you to work with the database. In database terms, the cursor is positioned at a particular location within a table or tables in a database.
- To get a cursor you need to call the cursor() method on the database object.
- To save your changes to the database, you must commit the transaction using **commit().**
- When you are done with the script, close the cursor and then the connection to free up the resources.
- The DB-API's include a defined set of exceptions.

Database Object Methods

Method	Description
close()	Closes the connection to the database.
commit()	Commits any pending transaction to the database.
cursor()	Returns a database cursor object through which queries can be executed.
rollback()	Rolls back any pending transaction to the state that existed before the transaction began.

► **Cursor object Attributes and Methods**

Note: c is the cursor object.

Syntax	Description
c.arraysize()	The readable/writable numbers of ours that fetchmany() will return if no size is specified.
c.close()	Close the cursor, c.
c.execute(sql,params)	Executes the SQL query on string sql, replacing each placeholder with the corresponding parameter from the param sequence or mapping
c.executemany(sql, seq-of-params)	Executes the SQL query once for each item in the seq-of-params sequence of sequences or mappings. This method should not be used for operations that create results set (such as SELECT statement)
c.fetchall()	Returns a sequence of all the rows that have not yet been fetched.
c.fetchmany(size)	Returns a sequence of rows(each row itself being a sequence); size defaults to c.arraysize
c.fetchone()	Returns the next row of the query result set as a sequence or None when the results are exhausted. Raises an exception if there is no result set.
c.rowcount()	The read only row count for the last operation (e.g. select, insert, update delete) or –1 if not available or applicable

Key Words

► **DB-API:** It is a specification for a common interface to relational databases.

► **Cursor:** It is a Python object that enables you to do the work with the database.

► **Result Set:** It is an object that is returned when a cursor object is used to query a table.

► **Database:** It is a collection of organised information that can easily be used, managed, updated and they are classified according to their organizational approach.

► **MySQLdb:** It is for connecting to a MySQL database server from Python.

■■