TABLE OF CONTENTS

7 .Data and DBMS Introduction:

7.1 Data and types of data

7.2 Database and types of database

7.3 DBMS&SQL

7.4 Installation of MYSQL

7.5 Creation of databases and tables insertion of values

8. SQL Commands

8.1 Sql select,where

8.2 Sql Datatypes

8.3 Date and Time

8.4 Characters

8.5 Server strings

8.6 Substring Datatype

9. DATABASES

9.1 Creating and Managing Databases

9.2 How to connect HTML to DBMS

9.3 How to store html data in databases

9.4 Creating tables in database

9.5 Joints

10. HTML INTRODUCTION

10.1 HTML Basics

10.2 HTML Tags

10.3 Images in webpage

10.4 Links in HTML

10.5 Creating tables and in webpage

10.6 Colors in HTML

11. CSS INTRODUCTION

11.1 What is CSS?

11.2 Different types of selectors

11.3 Fonts and Texts

11.4 Boxes

11.5 Colors in CSS

11.6 Style format in CSS

12. BOOTSTRAP INTRODUCTION

12.1 What is Boot Strap?

12.2 How to set up and use Boot strap?

12.3 Templates for typography forms buttonstables.

12.4 Navigation models,image carousels and many other

12.5 Bootstrap Buttons

12.6 Bootstrap Dropdowns

13. HTML TOPICS

13.1 Creation of Tables

13.2 Creating Iframes

13.3 Creating forms

13.4 Pattern Attributes

13.5 Inherritance

13.6 Method Overriding

14. PYTHON TOPICS

14.1 Operator Overloading

14.2 Method Overloading

14.3 Creating forms

14.4 Pattern Atrributes

14.5 Method Overriding

15. PYTHON TOPICS

15.1 Abstraction

15.2 Polymorphisom

15.3 Encapsulation

15.4 Exceptions

WEEKLY REPORT

WEEK -1 (From Dt.01/01/2024 to Dt.06/01/2024)

Detailed report:

INTRODUCTION TO PTHON PROGRAMMING:

- **Python** is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.
- Python is dynamically-typed and garbage-collected.
- It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming.
- It is often described as a "batteries included" language due to its comprehensive standard library.
- Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage
- collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward- compatible with earlier versions.

1.1 PYTHON AND INSTALLATION:

Installation steps:

Go to https://www.python.org/downloads. The most recent version of Python will always appear on the "Download" button near the top of the page.

If you want to use Python 3, see the "Installing Python 3" method.

Click Download Python <version>. If this does not start the download on the pop-up window to start it. immediately, click Save

Python 3.7 and newer will run on any Windows operating system except Windows XP. If you need to install Python 3 on XP, scroll down and

click Download next to the most recently-updated version of Python 3.4

Run the installer. You can do so by double-clicking python-<version>.exe in your Downloads folder.

Check the box next to "Add Python <version> to PATH." It's at the bottom of the window.
If you don't see this option, you'll need to complete this method after you finish installing Python.
Click Customize installation. It's the second blue link on the window.
Review the installation options and click Next. All Python's features are selected by default. Unless you have a specific need to skip installing any part of this package, just leave these settings alone.
Check the box next to "Install for all users."" If you're a system administrator, this option ensures that other users on this computer can use Python. This also changes the installation location to the Program Files\(x86)\ Python(version) instead of your personal library.
Click Install
Click Yes
It's at the bottom of the window.
ation is complete, you'll see a "Setup was successful" window— don't close it yet.
Click Disable path length limit. It's toward the bottom of the "Setup was successful" window. This final step ensures that Python (and other apps) can use paths more than 260 characters in length.
Click to confirm. Python is now installed and ready to use.

LIBRARIES:

In Python, libraries are defined as a package or collection of various modules, which includes various functions or methods in modules that are imported into the program to perform some task without writing the large code snippets in the program.

Some library functions:

- Standard
- NumPy
- SciPY
- Matplotlib
- Tkinter
- openCV
- pandas

Python uses:

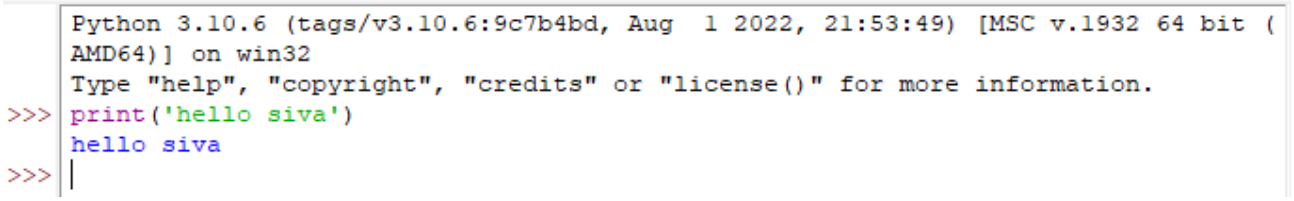
Python can be used to develop different applications like web applications, graphic user interface based applications, software development application, scientific and numeric applications, network programming, Games and 3D applications and other business applications. It makes an interactive interface and easy development of applications.

Examples: Web scraping applications, Game development etc.

Using the python interpreter IDLE:

- IDLE is the Python environment we will be using.
- IDLE stands for integrated development and learning environment.
- The IDLE shell window opens up. You can again type in print("hello siva") and so forth, and the shell will do the printing. As you can see, it's interactive. Python responds to every line of code you enter.
- Before running, IDLE prompts you to save the script as a file. Choose a name ending in .py ("hello siva.py") and save it on Desktop.

Example:



Salient features of python:

- Easy to read.
- Free and open source.
- Object oriented and procedure oriented.
- Support for GUI.
- Robust standard library.
- Easy to code.
- High – level language.
- Dynamically typed.
- Expressive

Basics syntax of python:

Syntax: Computers are machines that can interpret your instructions only if you type them in the exact format that the computers experts. This expected format – the spelling, formatting and grammar is called syntax.

Python uses:

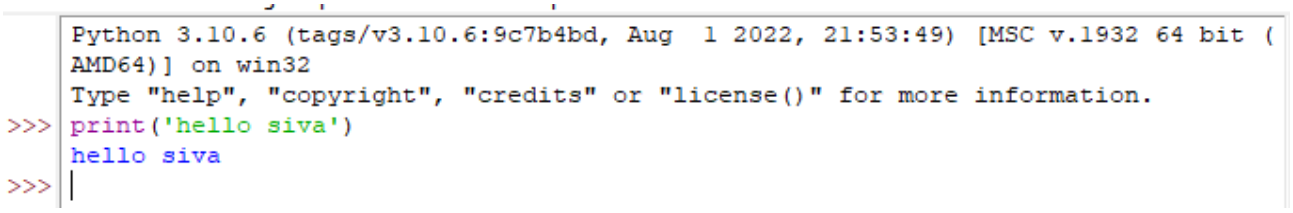
Python can be used to develop different applications like web applications, graphic user interface based applications, software development application, scientific and numeric applications, network programming, Games and 3D applications and other business applications. It makes an interactive interface and easy development of applications.

Examples: Web scraping applications, Game development etc.

1.2 Using the python interpreter IDLE:

- IDLE is the Python environment we will be using.
- IDLE stands for integrated development and learning environment.
- The IDLE shell window opens up. You can again type in print("hello siva") and so forth, and the shell will do the printing. As you can see, it's interactive. Python responds to every line of code you enter.
- Before running, IDLE prompts you to save the script as a file. Choose a name ending in .py ("hello siva.py") and save it on Desktop.

Example:



1.3 Salient features of python:

- Easy to read.
- Free and open source.
- Object oriented and procedure oriented.
- Support for GUI.
- Robust standard library.
- Easy to code.
- High – level language.
- Dynamically typed.
- Expressive

1.4 Basics syntax of python:

Syntax: Computers are machines that can interpret your instructions only if you type them in the exact format that the computers experts. This expected format – the spelling, formatting and grammar is called syntax.

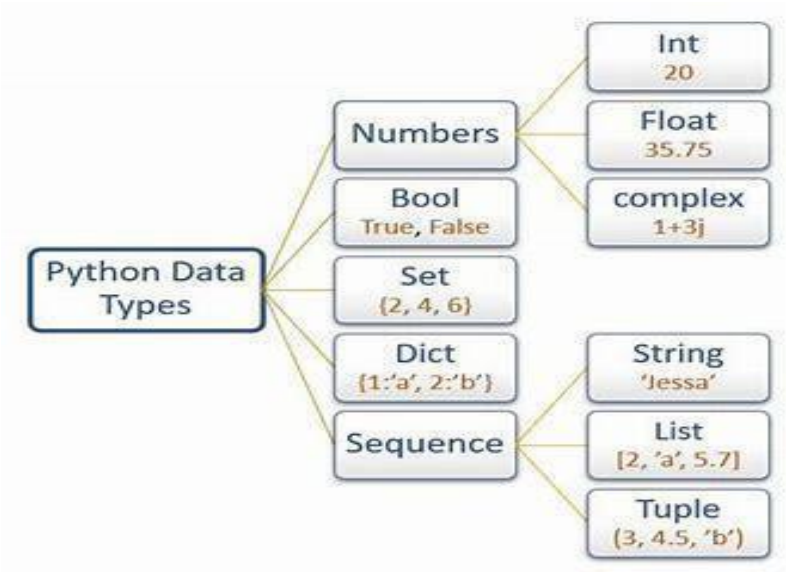
WEEKLY REPORT

WEEK -2 (From Dt.08/01/2024 to Dt.13/01/2024)

Using variables in Python:

2.1 Datatypes and Variables:

Datatype: Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data



Variables:

A Python variable is a name **given to a memory location**. It is the basic unit of storage in a program. The value stored in a variable can be changed during program execution.

Example:

```
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug  1 2022, 21:53:49) [MSC v.1932 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> var="siva kumar"
>>> print(var)
siva kumar
>>>
```

2.2 Numeric Data Types:

- Numerical data is a kind of **data expressed in numbers**. It is also sometimes referred to as quantitative data, which is always collected in numbers.

In Python, numeric data type represent the data which has numeric value.

Numeric value can be integer, floating number or even complex numbers. These values are defined as int, float and complex class in Python.

- **Integers**: This value is represented by int class. It contains positive or negative whole numbers (without fraction or decimal). In Python there is no limit to how long an integer value can be.
- **Float** :This value is represented by float class. It is a real number with floating point representation. It is specified by a decimal point. Optionally, the character e or E followed by a positive or negative integer may be appended to specify scientific notation.
- **Complex Numbers** : Complex number is represented by complex class. It is specified as (real part) + (imaginary part)j.
- For example – 2+3j

Example:

```
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug  1 2022, 21:53:49) [MSC v.1932 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> i=23
>>> j=3.0
>>> k=4+8j
>>> print(type(i))
<class 'int'>
>>> print(type(j))
<class 'float'>
>>> print(type(k))
<class 'complex'>
>>>
```

2.3 STRING DATA TYPES:

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

Example:

```
>>> s='hello how are you'
>>> print(s[1])
e
>>> print(s[2:6])
llo
>>> print(s[2:12:2])
lohwa
>>>
```

Sequence Data types:

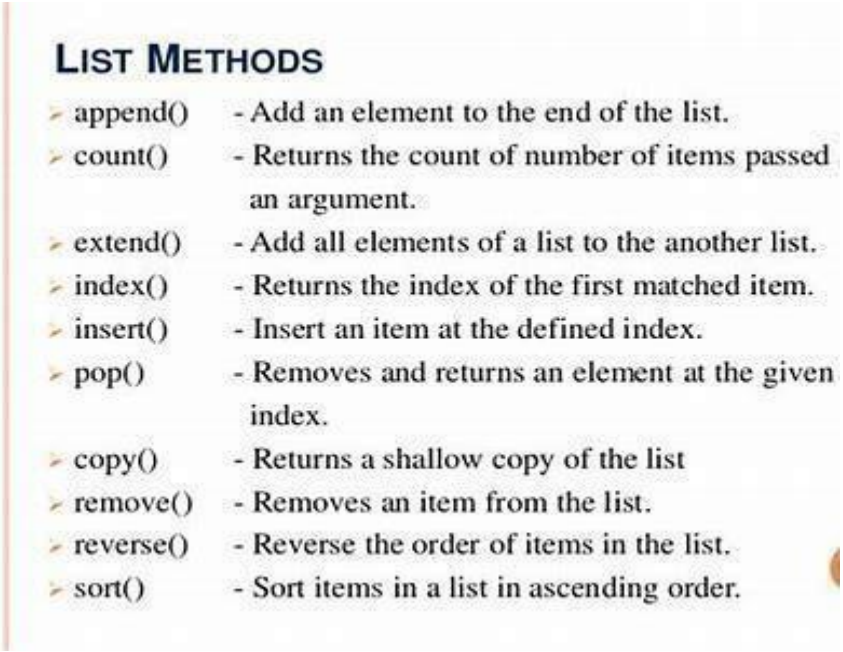
Sequence Data Types are used to store data in containers in the Python computer language. The different types of containers. Those are:

- List
- Tuple
- Strings

List:

The sequential data-type class includes the list data type. The list is the sole mutable data type in the sequential category. It can store any data type's values or components. Many procedures in the list can be changed and performed, such as append, remove, insert, extend, reverse, sorted, etc. We still have many more built-in functions to manipulate lists.

List functions



Example: List=["Tutorialspoint","is","the","best","platform","to","learn","new","skills"]
print(List)
print("Accessing element from the list") print(List[0])
print(List[3])
print("Accessing element from the list by using negative indexing") print(List[-2])
print(List[-3])

Strings:

The string values are stored using string data types. We can't manipulate the elements in a string because it's immutable. Strings have a lot of built-in functions, and we can use them to do a lot of things. The following are some of the built-in string functions count, isupper, islower,

Useful string functions & methods

Name	Purpose
len(s)	Calculate the length of the string s
+	Add two strings together
*	Repeat a string
s.find(x)	Find the first position of x in the string s
s.count(x)	Count the number of times x is in the string s
s.upper() s.lower()	Return a new string that is all uppercase or lowercase
s.replace(x, y)	Return a new string that has replaced the substring x with the new substring y
s.strip()	Return a new string with whitespace stripped from the ends
s.format()	Format a string's contents

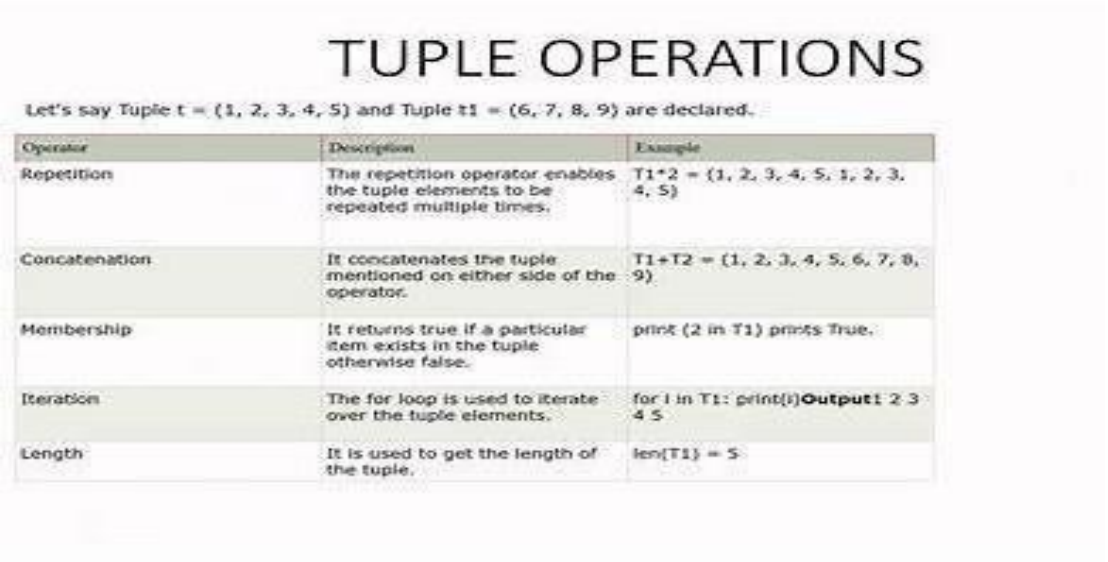
Example:

```
String="Tutorialspoint is the best platform to learn new skills" print(String)
print(type(String))
print("Accessing characters of a string:") print(String[6])
print(String[10])
print("Accessing characters of a string by using negative indexing")
print(String[-6])
print(String[-21])
```

```
=== RESTART: C:/Users/gayathri/AppData/Local/Programs/Python/Python310/ex8.py ==
Tutorialspoint is the best platform to learn new skills
<class 'str'>
Accessing characters of a string:
a
o
Accessing characters of a string by using negative indexing
s
m
```

Tuple:

Tuples are a data type that belongs to the sequence data type category. They're similar to lists in Python, but they have the property of being immutable. We can't change the elements of a tuple, but we can execute a ariety of actions on them such as count, index, type, etc.



Dictionary Data types:

Python dictionaries are **container data types**, like Python lists. Dictionaries use a key-value pair mapping, allowing you to retrieve a value by looking up its corresponding key. They are very similar to what, in other programming languages, is called an associative array.

Python Dictionary Methods	
Method	**Description**
clear()	Removes all the elements from the dictionary
copy()	Returns a copy of the dictionary
fromkeys()	Returns a dictionary with the specified keys and values
get()	Returns the value of the specified key
items()	Returns a list containing the a tuple for each key value pair
keys()	Returns a list containing the dictionary's keys
pop()	Removes the element with the specified key
popitem()	Removes the last inserted key-value pair
setdefault()	Returns the value of the specified key. If the key does not exist: insert the key, with the specifi
update()	Updates the dictionary with the specified key-value pairs
values()	Returns a list of all the values in the dictionary

Example:

```
dict={} dict['one']="This is
one" dict[2]="This is two"
tinydict={'name': 'john','code':6734, 'dept': 'sales'}
print(dict['one']) # Prints value for 'one' key print(dict[2]) #
Prints value for 2 key print(tinydict) # Prints complete
dictionary print(tinydict.keys()) # Prints all the keys
print(tinydict.values()) # Prints all the values
```

output:

```
This is one
This is two
{'name': 'john', 'code': 6734, 'dept': 'sales'}
dict_keys(['name', 'code', 'dept'])
dict_values(['john', 6734, 'sales'])
>>
```

WEEKLY REPORT

WEEK -3 (From Dt.15/01/2024 to Dt.20/01/2024)

3.1 Conditional statements:

A conditional statement in python, also called a **condition constructs**, is a statement that accommodates a condition inside itself. This condition is constructed using the bitwise, boolean, and comparison operators in Python.

- If Conditional Statement in Python.
- Else Statement in Python.
- Elif Conditional Statement in Python.
- Python Cascaded If Statements

If statement:

The ***"if "*** statement, as its name suggests evaluates the expression. Moreover, it executes the conditional block only when the statement is true. You can remember the statement as:

Syntax:

```
if (something is True):

{//statements}
```

Else statement:

whenever the "if " statement returns False and conditional block from "if " skips, the conditional block from "else " executes.

Syntax:

```
if (something is True):

{//statements}

else:
{//statements}
```

Else if Ladder:

Python elif (short for else if) is used to execute a continuous chain of **conditional logic ladder**. In elif, there are multiple conditions and the corresponding statement (s) as a ladder. Only one of the blocks gets executed when the corresponding boolean expression evaluates to true.

Syntax:

```
if(condition): #code
elif(condition): #code
elif(condition): #code
else:
    #code
```

Switch case:

- A switch statement is a multiway branch statement that compares the value of a variable to the values specified in case statements.
- Python language doesn't have a switch statement.
- Python uses dictionary mapping to implement Switch Case in Python

Syntax:

```
function(argument){ switch(argument) {
    case 0:
        return "This is Case Zero"; case
    1:
        return " This is Case One"; case
    2:
        return " This is Case Two "; default:
        return "nothing";
  };
};
```

3.2 Loops: A concept in Python programming package that allows repetition of certain steps, or printing or execution of the similar set of steps repetitively.

- While loop
- For loop
- Nested loop

while loop:

Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body.

for loop:

Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.

nested loops:

You can use one or more loop inside any another while, or for loop.

Example:

```
list1 = [5,10,15,20]

list2 = ['hi','hello','siva','good morning'] for x in

list1:

    for y in list2:
        print(x,y)
```

3.3 Functions: A function is a block of code which only runs when it is called.

- You can pass data, known as parameters, into a function.
- A function can return data as a result.

Creating a Function

- In Python a function is defined using the def keyword:

Ex:

```
def my_function():

  print("Hello from a python")

my_function()
```

Function with Arguments: Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

Ex:

```
def my_function(fname): print(fname + "

   Refsnes")

my_function("yerrakula")

my_function("pujari")

my_function("kandula")
```

3.4 Functions from Built in functions:

The Python built-in functions are defined as the functions whose functionality is pre-defined in Python. The python interpreter has several functions that are always present for use. These functions are known as Built-in Functions.

Built-in Functions				
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

Using construct modules and packages:

A module is a file with the extension.py that contains Python or C executable code. A module is made up of a number of Python statements and expressions. Modules allow us to use pre-defined variables, functions, and classes.

3.5 Creating a Module in Python

We can create a module by writing some code in a file and saving that file in a .py extension.
Let us create a module named pythongeeks.py.

Ex:

```
def display(): print("PythonGeeks")
if ___name____ == "___main___":
display()
```

Output: PythonGeeks

Importing a module in Python:

We can import a module by using the import keyword.

Example: import pythongeeks
pythongeeks.display()

output: Pythongreeks

packages: A package is a **collection of Python modules**, i.e., a package is a directory of Python modules containing an additional ____init____.py file. The _____init____.py distinguishes a package from a directory that just happens to contain a bunch of Python scripts.

Creating and Importing a Package:

To create a Python package, we need to create a directory with a _init___.py file and a module. Suppose we have created a package named website with the previously created module pythongeeks.py in it. We can import the website package by using the import keyword and a dot operator.

Ex:

```
import math
print(math.pi)
```

Output:

```
3.14159265358979
```

WEEKLY REPORT

WEEK -4 (From Dt.22/01/2024 to Dt.27/01/2024)

4.1 Overview of OOPS: Like other general-purpose programming languages, Python is also an object-oriented language since its beginning. It allows us to develop applications using an Object-Oriented approach. In Python, we can easily create and use classes and objects.

Major principles of object-oriented programming system are given below.

- Class
- Object
- Method
- Inheritance
- Polymorphism
- Data Abstraction
- Encapsulation

Class: The class can be defined as a collection of objects. It is a logical entity that has some specific attributes and methods. For example: if you have an employee class, then it should contain an attribute and method, i.e. an email id, name, age, salary, etc.

Object:

The object is an entity that has state and behavior. It may be any real-world object like the mouse, keyboard, chair, table, pen, etc.

Inheritance:

Inheritance is the most important aspect of object-oriented programming, which simulates the real-world concept of inheritance. It specifies that the child object acquires all the properties and behaviours of the parent object.

Polymorphism:

Polymorphism contains two words "poly" and "morphs". Poly means many, and morph means shape. By polymorphism, we understand that one task can be performed in different ways. For example - you have a class animal, and all animals speak. But they speak differently.

Encapsulation:

Encapsulation is also an essential aspect of object-oriented programming. It is used to restrict access to methods and variables. In encapsulation, code and data are wrapped together within a single unit from being modified by accident.

Data Abstraction:

Data abstraction and encapsulation both are often used as synonyms. Both are nearly synonyms because data abstraction is achieved through encapsulation.

Abstraction is used to hide internal details and show only functionalities. Abstracting something means to give names to things so that the name captures the core of what a function or a whole program does.

Define class:

Python is a completely object-oriented language. You have been working with classes and objects right from the beginning of these tutorials. Every element in a Python program is an object of a class.

4.2 Declaring a Class:

A class in Python can be declared using the class keyword.

```
class <ClassName>:
    <statement1>
    <statement2>
    .
    .
    <statementN>
```

Define objects:

An Object is an **instance of a Class**. A class is like a blueprint while an instance is a copy of the class with actual values. Python is object-oriented programming language that stresses on objects i.e. it mainly emphasizes functions.

Declaring objects:

I know how to create a new variables in python

```
obj = MyClass();
```

But this will actually define a new variable named obj other than just declaring a variable ofMyClass.

In Java we declare it using MyClass obj; and define MyClass obj = new MyClass();

4.3 Inheritance:

- One of the core concepts in object-oriented programming (OOP) languages is inheritance.
- It is a mechanism that allows you to create a hierarchy of classes that share a set of properties and methods by deriving a class from another class.
- Inheritance is the capability of one class to derive or inherit the properties from another class.

Syntax:

Class BaseClass:

{Body}

Class DerivedClass(BaseClass):

{Body}

Types of Inheritance:

- Single
- Multiple
- Multilevel
- Hierarichal
- Hybrid

Single inheritance: When a child class inherits from only one parent class, it is called single inheritance.

Multiple inheritances: When a child class inherits from multiple parent classes, it is called multiple inheritances.

Multilevel inheritance: When we have a child and grandchild relationship.

Hierarchical inheritance : More than one derived class are created from a single base.

Hybrid Inheritance: This form combines more than one form of inheritance. Basically, it is a blend of more than one type of inheritance.

Example:

```
class Base2(object):
    def __init__(self):
        self.str2 = "Geek2"
        print("Base2")
class Derived(Base1, Base2):
    def __init__(self):
        # Calling constructors of Base1 # and
```

```
        Base2       classes
        Base1._init_(self) Base2.
        init_____(self)
        print("Derived")
    def                 printStrs(self):
        print(self.str1, self.str2)
ob  =  Derived()
ob.printStrs()
```

output:

Base1

Base2

Derived

Geek1 Geek2

4.4 Magic methods:

Magic methods in Python are the special methods which add “magic” to your class. Magic methods are not meant to be invoked directly by you, but the invocation happens internally from the class on a certain action. For example, when you add two numbers using the + operator, internally, the __add__() method will be called.Python has many built-in magic methods to name some are:

- __init____
- __new____
- __del____
- __abs____
- __add____
- __sub____
- __mul____

```
Example: A._add_(5)
```

Output: 10

WEEKLY REPORT
WEEK -5 (From Dt.29/01/2024 to Dt.03/02/2024)

Detailed report:

5.1 Introduction to SQL:

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.
- What Can SQL do?
- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views
- SQL Commands:
- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.
- SQL can perform various tasks like create a table, add data to tables, drop
- the table, modify the table, set permission for users.

This is a SQL commands list that covers all the necessary actions with SQL databases. Each SQL command is provided with its definition, a code snippet that represents the correct syntax, and some have live code examples that you can try modifying to see the command in action.

Types of SQL commands:

SQL Joins:

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

Different Types of SQL JOINS

Here are the different types of the JOINs in SQL:

- (INNER) JOIN: Returns records that have matching values in both tables
- LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

SQL Keys:

In SQL, keys are the set of attributes that used to identify the specific row in a table and to find or create the relation between two or more tables i.e keys identify the rows by combining one or more columns.

The following are the list of SQL Keys:

- Primary Key
- Unique Key
- Candidate Key
- Alternate Key
- Composite Key
- Super Key
- Foreign Key

1. Primary Key:

Primary Key is a field that can be used to identify all the tuples uniquely in the database. Only one of the columns can be declared as a primary key. A Primary Key cannot have a NULL value.

2. Unique Key:

Unique Key can be a field or set of fields that can be used to uniquely identify the tuple from the database. One or more fields can be declared as a unique Key. The unique Key column can also hold the NULL value.

3. Candidate Key:

Candidate Key can be a column or group of columns that can qualify for the Unique Key. Every table has at least one Candidate Key. A table may have one or more Candidate Key. Each Candidate Key can work as a Primary Key if required in certain scenarios.

4. Alternate Key:

Alternate Key is that Key which can be used as a Primary Key if required. Alternate Key also qualifies to be a Primary Key but for the time being, It is not the Primary Key.

5. Composite Key:

Composite Key is also known as Compound Key / Concatenated Key. A group of the column in combination with each other can identify a row uniquely but a single column of that group doesn't promise to identify the row uniquely.

6. Super Key

Super Key is a combination of columns, each column of the table remains dependent on it. Super Key may have some more columns in the group which may or may not be necessary to identify the tuple uniquely from the table.

7. Foreign Key

A foreign key is a column which is known as Primary Key in the other table i.e. A Primary Key in a table can be referred to as a Foreign Key in another table.Foreign Key may have duplicate & NULL values.

5.2 Creating a SQLITE data base:

SQLite provides the create database functionality to users, in which that user can be able to create a database as per their requirement. SQLite gives you the alternative of making another database (or opening a current one) each time you start the order line utility. At the point when you use sqlite3 to begin the command line utility, you can alternatively affix a data set document name.

How to Create a Database in SQLite?

- First, we need to open the command-line interface that is the cmd from the start and type their cmd and open it.
- After that, cmd opens in the default folder of any machine.
- In the next step, find the desired SQLite location, which means it is located on the user machine.
- After that, select the specified file and navigate where the sqlite3.exe is placed by using the command-line tool.
- Once we are inside the SQLite, we can perform the create database operation and many more operations as per the requirement.

5.3 How To Use SQLite In Python.

Since SQLite database and its Python module have been installed with Python by default

WEEKLY REPORT

WEEK -6 (From Dt.05/02/2024 to Dt.10/02/2024)

6.1 GUI and Event driven programming:

Any program that uses GUI (**graphical user interface) such as Java application written for windows, is event driven**. Event describes the change in state of any object. For Example : Pressing a button, Entering a character in Textbox, Clicking or Dragging a mouse, etc. Events : An event is a change in state of an object.

6.2 QT designer:

Getting Started With Qt Designer

Qt Designer is a Qt tool that provides you with a what-you-see-is-what-you-get (WYSIWYG) user interface to create GUIs for your PyQt applications productively and efficiently. With this tool, you create GUIs by dragging and dropping QWidget objects on an empty form. After that, you can arrange them into a coherent GUI using different layout managers.

Qt Designer also allows you to preview your GUIs using different styles and resolutions, connect signals and slots, create menus and toolbars, and more.

6.3 Layout Manager:

Layout managers are also called as **geometry managers**. They are used for positioning,arranging and registering widgets on tkinter window.

They are used for positioning,arranging and registering widgets on tkinter window.

Python provides three layout/ geometry managers.

1. pack()
2. grid()
3. place()

pack()

The pack method is used to organize widget in to cavity on parent window.

Syntax

w.pack(options)

grid()

The grid method is used to organize widget in to grid / table like structure.

Syntax

w.grid(options)

place()

The place method is used to organize widget in to parent window placing them in specific position.

In absolute positioning we specify the position and size of each widget in pixels. The size and position of widget do not change if we resize the window

Syntax

w.place(options)

6.4 Using common widgets:

Widgets are small parts on web applications that allow user input. The widgets are eventful objects of Python that have a browser representation, often as control such as a slider, textbox, etc. Interactive GUIs for Python notebook are created using these Widgets. Synchronize stateless and stateful information between JavaScript Python and is done using widgets. Widget input allows us to add parameters and dashboards to the notebooks.

Syntax:

```
import widget_name Ex:
import Ipywidgets as wg
    import Ipython.display
    name = wg.Text (value = 'Name')
    age = wg.IntSlider (description = 'Age: ')
    display (name, age)
    print (name.value + ' is already '+ str (age.value) + ' now ')
```

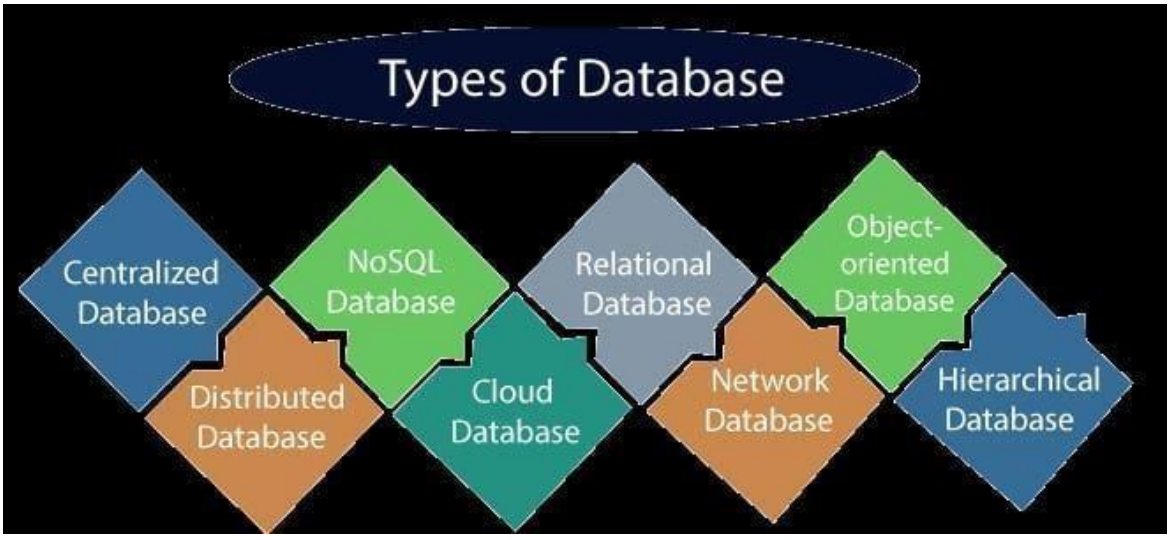
7.1 Data:

Data in SQL refers to information that is organised and structured and kept in a relational database.

Types of data: There are three main data types: string, numeric, and date and time.

7.2 Database:

SQL database or relational database is a collection of highly structured tables, wherein each row reflects a data entity, and every column defines a specific information field. Relational databases are built using the structured query language (SQL) to create, store, update, and retrieve data. Therefore, SQL is the underlying programming language for all relational database management systems (RDBMS) such as MySQL, Oracle, and Sybase, among others.



Types of Databases:

There are various other widely used databases in addition to the two primary types, which are as follows:

Hierarchical Databases

A hierarchical database is a type of database that uses a hierarchical model to organize data. In a hierarchical database, data is organized into a tree-like structure, with each record represented as a node in the tree. Each node is connected to one or more child nodes, and each child node can have its own set of child nodes, creating a hierarchy of nodes.

In a hierarchical database, each node in the hierarchy can have only one parent node, and the relationship between nodes is one-to-many. This means that a parent node can have multiple child nodes, but a child node can only have one parent node.

Network Database

A network database uses a network model to organize data. In a network database, data is organized into a series of interconnected records, with each record representing an entity and the relationships between the entities represented as lines connecting the records.

Each record in this type of database can have multiple parent and child records, creating a complex web of relationships between the data. This allows for a more flexible and expressive data model than hierarchical databases, which have a more rigid one-to-many relationship between parent and child nodes.

Object-Oriented Databases

In an object-oriented database, data is organized into objects, which are self-contained entities that contain both data and the methods that operate on that data. Object-oriented databases are designed to support the creation and management of complex data structures, and they are often used in applications that require the manipulation of large amounts of structured and semi-structured data.

Relational Databases

These are the most widely used type of databases, and they store data in tables that are related to each other through common keys or indexes. Examples of relational databases include MySQL, Oracle, and Microsoft SQL Server.

NoSQL Databases

These databases are designed to handle large amounts of unstructured or semi-structured data, and they do not use the traditional table-based relational database model. Instead, they use a variety of data models, such as key-value pairs, documents, and graphs, to store data. Examples of NoSQL databases include MongoDB, Cassandra, and Couchbase.

Centralized Database

A centralized database is a database that is stored on a central server and can be accessed by multiple users over a network. The central server acts as a hub for the database, and all users access the same copy of the database. This database is commonly used in organizations to store and manage data that is shared by multiple users or departments.

Distributed Database

A distributed database is stored and managed across multiple servers, rather than on a single central server. It is designed to provide faster access to data and to improve the scalability and reliability of the database. In a distributed database, data is divided into smaller chunks and stored on multiple servers, with each server responsible for storing and managing a portion of the data.

Cloud Database

A cloud database is stored and managed on a cloud computing platform, rather than on a local server or device. Cloud databases are accessed over the internet and can be used by multiple users or applications, providing a flexible and scalable way to store and manage data. Cloud databases can be used for a wide range of applications, including web and mobile applications, data warehousing, and more.

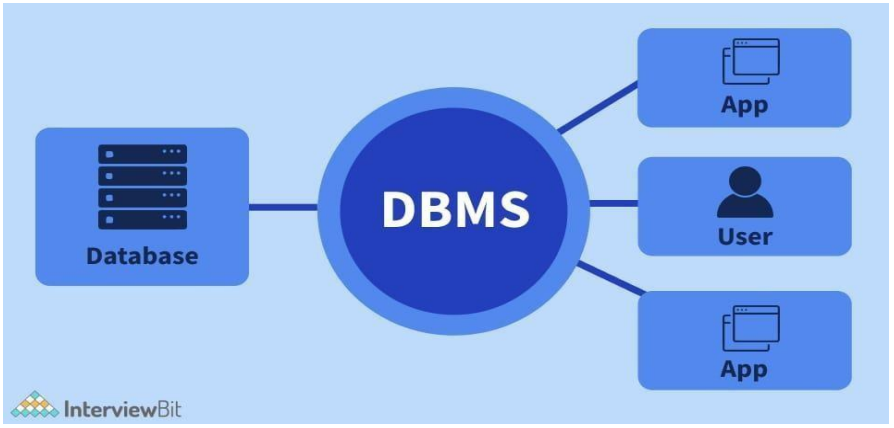
Personal Database

A personal database is designed to store and manage data for a single individual or small group of users. They are typically smaller in scale and scope than enterprise databases, which are used by larger organizations to store and manage data for a large number of users. You can use personal databases to store and manage a wide range of data, including financial records, contact information, personal notes, and more.

7.3 What is DBMS?

Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data. A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.

SQL Introduction:



SQL is a standard language for accessing and manipulating databases.

SQL stands for Structured Query Language

SQL lets you access and manipulate databases

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

What Can SQL do?

SQL can execute queries against a database

SQL can retrieve data from a database

SQL can insert records in a database

SQL can update records in a database

7.4 Installation of MYSQL:

Install MySQL

After downloading, unzip it, and double click the MSI installer .exe file.

Then follow the steps below:

1. "Choosing a Setup Type" screen: Choose "Full" setup type. This installs all MySQL products and features. Then click the "Next" button to continue.

2. "Check Requirements" screen: The installer checks if your pc has the requirements needed. If there is some failing requirements, click on each item to try to resolve them by clicking on the Execute button that will install all requirements automatically. Click "Next".

3. "Installation" screen: See what products that will be installed. Click "Execute" to download and install the Products. After finishing the installation, click "Next".

4. "Product Configuration" screen: See what products that will be configured. Click the "MySQL Server 8.0.23" option to configure the MySQL Server. Click the "Next" button. Choose the "Standalone MySQL Server/Classic MySQL Replication" option and click on the "Next" button. In page "Type and Networking" set Config Type to "Development Computer" and "Connectivity" to "TCP/IP" and "Port" to "3006". Then, click the "Next" button.

5. "Authentication Method" screen: Choose "Use Strong Password Encryption for Authentication". Click "Next".

6. "Accounts and Roles" screen: Set a password for the root account. Click "Next".

7. "Windows Service" screen: Here, you configure the Windows Service to start the server. Keep the default setup, then click "Next".

8. "Apply Configuration" screen: Click the "Execute" button to apply the Server configuration. After finishing, click the "Finish" button.

9. "Product Configuration" screen: See that the Product Configuration is completed. Keep the default setting

and click on the "Next" and "Finish" button to complete the MySQL package installation.

10. In the next screen, you can choose to configure the Router. Click on "Next", "Finish" and then click the "Next" button.

11. "Connect To Server" screen: Type in the root password (from step 6). Click the "Check" button to check if the connection is successful or not. Click on the "Next" button.

12. "Apply Configuration" screen: Select the options and click the "Execute" button. After finishing, click the "Finish" button.

13. "Installation Complete" screen: The installation is complete. Click the "Finish" button**.**

7.4 Installation of Mysql:

1.To install Oracle database for Windows operating system:

2 Go to Oracle Database Software Downloads in your browser.

3 Download the 64-bit .zip file. Select the.zip file and right click to select Extract All. Extract both the .zip files to the same folder.

4 Don't extract the archive using unzip command, which may result in an unsuccessful run of the setup.exe.

5 Run the setup.exe and select the installation options according to your database and Windows user requirements.

6 In the Specify Database Identifiers screen of the installation process, enter the Global database name (for example, amc2) and the Oracle system identifier, SID (for example, amc2). Don't select the check box for the option Create as Container database. The Install button gets enabled.

7 Click Install to install the product.

8 Oracle Database is installed on Windows.

9 Start the SQL Plus application. From the command-line, enter the command SQLPLUS to start SQL Plus.

10 From Windows Start, click Programs, Oracle-Ora Home Name, Application Development, and SQL Plus.

7.5 Creation of Databases, Tables and Values:

The SQL CREATE DATABASE Statement

The CREATE DATABASE statement is used to create a new SQL database.

Syntax

CREATE DATABASE databasename;

The syntax of the CREATE TABLE statement

The basic syntax we use to create a new table on SQL Server is:

```
CREATE TABLE [database_name.][schema_name.]table_name (
    column_name1 data_type [NULL | NOT NULL],
    column_name2 data_type [NULL | NOT NULL],
    column_name3 data_type [NULL | NOT NULL],
    ...,
);
```

INSERTION:

The SQL INSERT INTO Statement

The INSERT INTO statement is used to insert new records in a table.

INSERT INTO Syntax

It is possible to write the INSERT INTO statement in two ways:

1. Specify both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2, column3, ...)VALUES (value1, value2, value3, ...);
```

2. If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. Here, the INSERT INTO syntax would be as follows:

```
INSERT INTO table_name VALUES (value1, value2, value3, ...);
```

8.1 SQL - [SELECT, WHERE, PROJECTION AND SELECTION]

The SQL SELECT Statement

The SELECT statement is used to select data from a database.

Example:

Return data from the Customers table:

SELECT CustomerName, City FROM Customers;

Syntax

SELECT column1, column2, ...

FROM table_name;

The SQL WHERE Clause

The WHERE clause is used to filter records.

It is used to extract only those records that fulfill a specified condition.

Example:

Select all customers from Mexico:

SELECT * FROM Customers WHERE Country='Mexico';

Syntax

SELECT column1, column2, ...

FROM table_name

WHERE condition;

Note: The WHERE clause is not only used in SELECT statements, it is also used in UPDATE, DELETE, etc.!

1. Selection :

This operation chooses the subset of tuples from the relation that satisfies the given condition mentioned in the syntax of selection.

Notation –σc (R)

Here, ‘c’ is selection condition and ‘σ (sigma)’ is used to denote Select Operator.

2. Projection :

This operation selects certain required attributes, while discarding other attributes.

Notation –

πA (R)
where 'A' is the attribute list, it is the desired set of attributes from the attributes of relation(R),
symbol 'π(pi)' is used to denote the Project operator,
R is generally a relational algebra expression, which results in a relation

8.2 SQL Data Types

An SQL developer must be aware of what type of data will be stored inside each column while creating a table. The data type guideline for SQL is to understand what type of data is expected inside each column and it also identifies how SQL will interact with the stored data.For every database, data types are primarily classified into three categories.

- Numeric Datatypes
- Date and Time Database
- String Database
- Numeric Data Types in MYSQL
- Exact Numeric Datatype

There are nine subtypes which are given below in the table. The table contains the range of data in a

Data Type	From	To
BigInt	-2^{63} (-9,223,372,036,854,775,808)	2^{63} -1 (9,223,372,036,854,775,807)
Int	-2^{31} (-2,147,483,648)	2^{31}-1 (2,147,483,647)
smallint	-2^{15} (-32,768)	2^{15}-1 (32,767)
tinyint	0	2^{8}-1 (255)
bit	0	1
decimal	-10^{38}+1	10^{38}-1
numeric	-10^{38}+1	10^{38}-1
money	-922,337,203,685,477.5808	922,337,203,685,477.5807
smallmoney	-214,748.3648	214,748.3647

8.3 Server Date and Time Data Type in SQL.The details are given in the below table.

Data Type	Description
DATE	A data type is used to store the data of date in a record
TIME	A data type is used to store the data of time in a record
DATETIME	A data type is used to store both the data,date, and time in the record.

8.4 String Data Types in MYSQL Character String DatatypeThe subtypes are given in below table –

Data Type	Description
char	The maximum length of 8000 characters.(Fixed-Length non-Unicode Characters)
varchar	The maximum length of 8000 characters.(Variable-Length non-Unicode Characters)
varchar(max)	The maximum length of 231 characters(SQL Server 2005 only).(Variable Length non-Unicode data)
text	The maximum length of 2,127,483,647 characters(Variable Length non-Unicode data)

8.5 Server String Data Type in SQL.There are four subtypes of this datatype which are given below:

Datatypes	Description
Binary	The maximum length of 8000 bytes(Fixed-Length binary data)
varbinary	The maximum length of 8000 bytes(Variable Length binary data)
varbinary(max)	The maximum length of 231 bytes(SQL Server 2005 only).(Variable Length binary data)
text	Maximum Length of 2,147,483,647 bytes(Variable Length binary data)

8.6 Unicode Character String DatatypeThe details are given in below table –

Data Type	Description
nchar	The maximum length of 4000 characters(Fixed-Length Unicode Characters)
Nvarchar	The maximum length of 4000 characters.(Variable-Length Unicode Characters)
nvarchar(max)	The maximum length of 231 characters(SQL Server 2005 only).(Variable Length Unicode data)

WEEKLY REPORT

WEEK – 9 (From Dt…20/02/2024. to Dt…29/02/2024.)

9.1 Creating a Database

The CREATE DATABASE command creates a new database. For example:

To create a database, you must have privileges to create a database or be a Greenplum Database superuser. If you do not have the correct privileges, you cannot create a database. Contact your Greenplum Database administrator to either give you the necessary privilege or to create a database for you.

You can also use the client program createdb to create a database. For example, running the following command in a command line terminal connects to Greenplum Database using the provided host name and port and creates a database
named mydatabase:

The host name and port must match the host name and port of the installed Greenplum Database system.

Some objects, such as roles, are shared by all the databases in a Greenplum Database system. Other objects, such as tables that you create, are known only in the database in which you create them.

Warning: The CREATE DATABASE command is not transactional.

Cloning a Database

By default, a new database is created by cloning the standard system database template, template1. Any database can be used as a template when creating a new database, thereby providing the capability to 'clone' or copy an existing database and all objects and data within that database. For example:

9.2 Creating a Database with a Different Owner

Viewing the List of Databases
If you are working in the psql client program, you can use the \l meta-command to show the list of databases and templates in your Greenplum Database system. If using another client program and you are a superuser, you can query the list of databases from the pg_database system catalog table. For example:

Altering a Database:

The ALTER DATABASE command changes database attributes such as owner, name, or default configuration attributes. For example, the following command alters a database by setting its default schema search path (the search_path configuration parameter):

To alter a database, you must be the owner of the database or a superuser.

Dropping a Database:

The DROP DATABASE command drops (or deletes) a database. It removes the system catalog entries for the database and deletes the database directory on disk that contains the data. You must be the database owner or a superuser to drop a database, and you cannot drop a database while you or anyone else is connected to it. Connect to postgres (or another database) before dropping a database. For example:

You can also use the client program drop db to drop a database. For example, the following command connects to Greenplum Database using the provided host name and port and drops the database my database:

Warning: Dropping a database cannot be undone.

The DROP DATABASE command is not transactional.

The SQL CREATE TABLE Statement

The CREATE TABLE statement is used to create a new table in a database

Syntax

```
CREATE TABLE table_name (
column1 datatype, column2 datatype, column3 datatype,
....
);
```

The column parameters specify the names of the columns of the table.

The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.

9.4 SQL CREATE TABLE Example:

The following example creates a table called "Persons" that contains five columns: PersonID, LastName, FirstName, Address, and City:

Example

```
CREATE TABLE Persons ( PersonID int,
LastName varchar(255), FirstName varchar(255), Address varchar(255), City varchar(255)
);
```

9.5 Joins:

SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are as follows:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

WEEKLY REPORT

WEEK – 10 (From Dt…01/03/24.. to Dt…15/03/24…)

10.1 HTML Basics and Tags:

Headings may be one of the easiest codes to learn and considering how crucial they areto your SEO, it's a good thing. There are six different types, as seen below. To create a heading, simply wrap your text in the heading tags of your choice.

1.Paragraphs

What would a nice heading be without a paragraph to elaborate on the message? To geta paragraph like the one you're reading now, simply wrap your text in tags like the example below, and don't forget to close it with a tag!

Hey, I'm a paragraph!

Hey, I'm a paragraph!

10.3 Links

Inbound marketing is nothing without linking your already-great content to other relevant articles and website pieces. Try linking a word or phrase in your paragraph byusing the following <a> code:

Ex: *Let's visit IMPACT's awesome website!*

10.4 Images:

This is a function one. Images make everything better, and they make your content a lotmore appealing to the reader. Insert an image like this:

Ex: *<img class="lazy" data-src="https://cdn2.hubspot.net/hubfs/145335/Cute-Puppy.jpg" alt="Cute-Puppy" style="width:500px;height:333px;">*

The image tag is empty because it only contains attributes, so it doesn't need to beclosed. The attributes listed above include "src" or image URL.

4. Line break

A line break is also an empty element, so it doesn't need to be closed. A line break isbasically an intentional space between two lines of text, created with
.

***Ex*:** Place a line break underneath this sentence.s

Place a line break above this sentence.

O/p: Place a line break underneath this sentence

Place a line break above this sentence.

5. Bold and strong:

To make something bold, there are two code elements that work. However, my developers tell me that is used much more than <b>. Don't forget to close the tag!

***Ex*:**Bold a whole sentence!Or only
bold one word!

Bold a whole sentence!
Or only **bold** one word!

6. Italic and emphasized:

Italic and emphasized text are similar to bold and strong text. There are two code elements, but one is used more than the other. In this case, <i> will work, but ismore commonly used.

em>This sentence is super fancy.This

sentence is super fancy.

7. Underlined:

Bold, italicized, and now underlined. This one is just as easy as the other two. Just wrapthe text you want underlined in <u> tags, like this.

<u>Look, we can underline!</u>

Look, we can underline!

8. Ordered lists:

There is a difference between an ordered list and an unordered list. The ordered list contains numbers, while the unordered list contains bullet points. They both follow thesame structure, but one letter changes.

Here is the code sentence for an ordered list. The <ol> is the entire "ordered list," whilethe <li> is a "list item." You can include as many list items as you need.

  - This is the first item - This is the second item. - This is the third item. 	1. This is the first item. 2. This is the second item. 3. This is the third item.

9. Unordered lists:

Only one letter change is needed to switch from an ordered list to an unordered

  - This is the first item - This is the second item. - This is the third item. 	1. This is the first item. 2. This is the second item. 3. This is the third item.

10. Upperscript:

To insert a superscript format within your text, wrap the text you want to appearsuperscripted in ^{tags.} You'll end up with something like this.

Ex:*Trademarks should be written in superscript^{TM}.*

Trademarks should be written in superscript^{TM}.

11. Subscript:

If you know how to superscript, you should know how to subscript. Just use _{tags} so you get text like this.

Ex:Sometimes, citations are written in subscript.

Sometimes, $_{\text{citations}}$ are written in subscript.

12. Horizontal line:

Want to break up sections of a page or article? Try a horizontal line! Just use the emptyelement (no need to close it), <hr>.

Ex:Insert a horizontal line between me and sentence 2.
<hr>
Hi, I'm sentence 2.

Insert a horizontal line between me and sentence 2

Hi, I'm sentence 2.

13. Marked or highlighted text:

I bet you didn't know that you could highlight text through an HTML code, did you? It'sso cool and so easy. Wrap the text to be highlighted in <mark> tags </mark> so that youget a cool highlighted feature.

Only highlight the <mark>most important notes</mark>.Only
highlight the most important notes.

14. Deleted (cross-through) text:

If you want to display the cross-through effect on your text (maybe you created a tasklist and want to cross out each one as you go), there's a code for that. Try crossing outfull sentences or even just one word by using <del> tags. </del>

***Ex*:**~~Feed the dog.~~
~~Write my blog article.~~
Make dinner.

Make dinner.

15. Short and long quotations:

It's actually called a blockquote or a long quote. You can see the difference between along quote and a short quote (normal quotation marks) below.

<blockquote>All of this text will be in a blockquote like the rest of the examples.</blockquote>
<q>I'm quoting this because I'm saying it out loud.</q>

All of this text will be in a blockquote like the rest of the examples.
I'm quoting this because I'm saying it out loud.

16. Setting a specific font:

The next few are going to get a bit trickier, so try to stay with me. Now that you knowhow to create a heading, a paragraph, and stylized text, it'll be useful for you to know that you can easily change the font using the element "font-family."

<h4 style="font-family:Georgia;">I want to change this header to Georgia font.</h4>

* I want to change this paragraph to Verdanafont. *

I want to change this header to Georgia font.

I want to change this paragraph to Verdana font.

17. Setting a specific text color:

This one uses the same code type as the previous example, but instead of using font- family, you use "color."

You can experiment with actual colors (blue, red, orange, etc.),or you can also insert hex colors to customize your text to your brand.

Ex: The sky is really blue today.
The fire is a reddish-orange.

The sky is really blue today.

The fire is a reddish-orange.

18. Setting a specific text size:

Again, this one uses the same basic code logic, but uses the element "font-size." Putyour font sizes in pixels, or px.

Ex: Make this paragraph size 36 font.
Make this paragraph size 12 font. Make

this paragraph size 36 font.

Make this paragraph size 12 font.

19. Setting a specific text alignment:

Left, centered, right, or justified. How do you like your text? Make it any way you'd likewith "text-align."

This paragraph should be centered.
This paragraph should be right aligned.

This paragraph should be centered.

This paragraph should be right aligned.

10.5 Tables:

I'll show you how to create a simple table below.To make it easier to understand, <tr>stands for table row, while <td> stands for table data. Keep in mind that you can changethe font, text size, text color, text alignment, and more.

11.1 What is CSS?

Like HTML, CSS is not a programming language. It's not a markup language either. **CSS is a style sheet language.** CSS is what you use to selectively style HTMLelements. For example, this CSS selects paragraph text, setting the color to red:

```
p {
  color: red;
}
```

11.2 Different types of selectors:

There are many different types of selectors. The examples above use **element selectors**, which select all elements of a given type. But we can make more specificselections as well. Here are some of the more common types of selectors:

Selector name	**What does it select**	**Example**
Element selector (sometimes called a tag or type selector)	All HTML elements of the specified type.	p selects <p>
ID selector	The element on the page with the specified ID. On a given HTML page, each id value should be unique.	#my-id selects <p id="my-id"> or <a id="my-id">
Class selector	The element(s) on the page with the specified class. Multiple instances of the same class can appear on a page.	.my-class selects <p class="my-class"> and <aclass="my-class">
Attribute selector	The element(s) on the page with the specified attribute.	img[src] selects <img src="myimage.png"> butnot <img>
Pseudo-class selector	The specified element(s), but only when in the specified state. (For example, when a cursor hovers overa link.)	a:hover selects <a>, but only when the mouse pointer is hovering overthe link.

WEEKLY REPORT

WEEK – 12 (From Dt…04/0/24 to Dt 15/04/24….)

12.1 What is Bootstrap?

- Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
- It is absolutely free to download and use.
- It is a front-end framework used for easier and faster web development.
- It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
- Setting Up Bootstrap on Your Server:

This tutorial will take you through setting up Bootstrap on your server. Bootstrap is a free, front-end framework for designing websites and web applications. It's very popular for website design because of its many easily added components and responsiveness.

Keep in mind that this tutorial goes over just one of multiple types of Bootstrap installations. We will be using the standard Bootstrap install. This is best for Bootstrap beginners. The other installations require knowledge of CSS preprocessors.

Templates for typographyforms buttons tables:

Create a logo that depicts your brand with help from our professionally designed library. Canva has a wide variety of creative typography logo designs ranging from classic to modern layouts. There are also colorful and monochromatic designs. Once you've settled on a template, you can personalize it on our easy-to-use editor.

12.2 Bootstrap button:

Bootstrap's .button styles can be applied to other elements, such as <label>s, to provide checkbox or radio style button toggling. Add data-toggle="buttons" to a.btn-group containing those modified buttons to enable their toggling behavior via JavaScript and add .btn-group-toggle to style the <input> s within your bootstrap.

Examples:

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control.

Primary Secondary Success Danger Warning Info Light Dark Link Copy

<button type="button" class="btn btn-primary">Primary</button>

```
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-link">Link</button>
```

Outline buttons:

In need of a button, but not the hefty background colors they bring? Replace the default modifier classes with the .btn-outline-* ones to remove all background images and colors on any button.

Checkbox and radio buttons

Bootstrap's .button styles can be applied to other elements, such as <label>s, to provide checkbox or radio style button toggling. Add data-toggle="buttons" to a .btn- group containing those modified buttons to enable their toggling behavior via JavaScript and add .btn-group-toggle to style the <input>s within your buttons. Note that you can create single input-powered buttons or groups of them.

The checked state for these buttons is only updated via click event on the button. If you use another method to update the input—e.g., with <input type="reset"> or by manually applying the input's checked property—you'll need to toggle .active on the <label> manually.

Note that pre-checked buttons require you to manually add the .active class to the input's <label>.

WEEKLY REPORT

WEEK – 13 (From Dt…16/04/24 to Dt 23/04/24….)

13.1 Creation of Tables using HTML:

What is HTML Tables?

An HTML Table is an arrangement of data in rows and columns in tabular format. Tables are useful for various tasks, such as presenting text information and numerical data. A table is a useful tool for quickly and easily finding connections between different types of data. Tables are also used to create databases.

Example:

```
<!-- index.html -->
<!DOCTYPE html>
<html>
<body>
   <table>
      <tr>
         <th>Firstname</th>
         <th>Lastname</th>
         <th>Age</th>
      </tr>
      <tr>
         <td>Priya</td>
         <td>Sharma</td>
         <td>24</td>
      </tr>
      <tr>
         <td>Arun</td>
         <td>Singh</td>
         <td>32</td>
      </tr>
      <tr>
         <td>Sam</td>
         <td>Watson</td>
         <td>41</td>
      </tr>
   </table>
</body>
</html>
```

Output:

Firstname	Lastname	Age
Priya	Sharma	24
Arun	Singh	32
Sam	Watson	41

Defining Tables in HTML

An HTML table is defined with the “table” tag. Each table row is defined with the “tr” tag. A table header is defined with the “th” tag. By default, table headings are bold and centered. A table data/cell is defined with the “td” tag.

Table Cells

Table Cell are the building blocks for defining the Table. It is denoted with <td> as a start tag & </td> as a end tag.

Syntax

</td> Content...</td>

Table Rows

The rows can be formed with the help of combination of Table Cells. It is denoted by <tr> and </tr> tag as a start & end tags.

Syntax

</tr> Content...</tr>

Table Headers

The Headers are generally use to provide the Heading. The Table Headers can also be used to add the heading to the Table. This contains the <th> & </th> tags.

Syntax

</th> Content...</th>

13.2 Creating iframes:

An HTML iframe is used to display a web page within a web page.

HTML Iframe Syntax

The HTML <iframe> tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

Syntax

<iframe src="url" title="description"></iframe>

Tip: It is a good practice to always include a title attribute for the <iframe>. This is used by screen readers to read out what the content of the iframe is.

Iframe - Set Height and Width

Use the height and width attributes to specify the size of the iframe.

The height and width are specified in pixels by default

13Forms:

HTML Forms employ the <form> element to gather input data with interactive controls. It encompasses various input types such as text, numbers, email, password, checkboxes, radio buttons, and submit buttons. Essentially, it's a container for diverse input elements facilitating user interaction.

Syntax:

```
<form>
    <!--form elements-->
</form>
```

HTML Forms Example

1. Basic HTML Forms Example:

Example: This HTML forms collects the user personal information such as username and password with the button to submit the form.

```
 <!DOCTYPE html>
<html lang="en">
<head>
<title>Html Forms</title>
</head>
<body>
  <h2>HTML Forms</h2>
  <form>
    <label for="username">Username:</label><br>
    <input type="text" id="username" name="username"><br><br>

    <label for="password">Password:</label><br>
    <input type="password" id="password" name="password"><br><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

Output:

HTML Forms Example:

Example: This HTML form collects user personal information including name, email, password, gender, date of birth, and address. It features proper styling for input fields and submission button.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Form</title>
    <style>
      body {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        margin: 0;
        background-color: #f0f0f0;
      }

      form {
        width: 400px;
        background-color: #fff;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 0 10px
          rgba(0, 0, 0, 0.1);
      }

      fieldset {
        border: 1px solid black;
        padding: 10px;
        margin: 0;
      }

      legend {
        font-weight: bold;
        margin-bottom: 10px;
      }
```

```
        label {
            display: block;
            margin-bottom: 5px;
        }

        input[type="text"],
        input[type="email"],
        input[type="password"],
        textarea,
        input[type="date"] {
            width: calc(100% - 20px);
            padding: 8px;
            margin-bottom: 10px;
            box-sizing: border-box;
            border: 1px solid #ccc;
            border-radius: 4px;
        }

        input[type="radio"] {
            margin-left: 20px;
        }

        input[type="submit"] {
            padding: 10px 20px;

            border-radius: 5px;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <form>
        <fieldset>
            <legend>
                User personal information
            </legend>
            <label>Enter your full name</label>
```

```
<input type="text" name="name" />
<label>Enter your email</label>
<input
   type="email"
   name="email"
/>
<label>Enter your password</label>
<input
   type="password"
   name="pass"
/>
<label
   >Confirm your password</label
>
<input
   type="password"
   name="confirmPass"
/>
<label>Enter your gender</label>
<input
   type="radio"
   name="gender"
   value="male"
/>Male
<input
   type="radio"
   name="gender"
   value="female"
/>Female
<input
   type="radio"
   name="gender"
   value="others"
/>Others
<label
   >Enter your Date of
   Birth</label
```

```
            >
            <input type="date" name="dob" />
            <label>Enter your Address:</label>
            <textarea
               name="address"
            ></textarea>
            <input
               type="submit"
               value="submit"
            />
         </fieldset>
      </form>
   </body>
</html>
```

Output:

13.4 Pattern Attribute:

This attribute is used to specify the regular expression on which the input element value is checked. This attribute works with the following input types: text, password, date, search, email, etc. Use the Global title attribute to describe the pattern for helping the user.

WEEKLY REPORT

WEEK – 14 (From Dt…24/04/24 to Dt 30/04/24….)

14.1 Operator Overloading:

Operator Overloading means giving extended meaning beyond their predefined operational meaning. For example operator + is used to add two integers as well as join two strings and merge two lists. It is achievable because '+' operator is overloaded by int class and str class. You might have noticed that the same built-in operator or function shows different behavior for objects of different classes, this is called Operator Overloading.

How to overload the operators in Python?

Consider that we have two objects which are a physical representation of a class (user-defined data type) and we have to add two objects with binary '+' operator it throws an error, because compiler don't know how to add two objects. So we define a method for an operator and that process is called operator overloading. We can overload all existing operators but we can't create a new operator. To perform operator overloading, Python provides some special function or magic function that is automatically invoked when it is associated with that particular operator. For example, when we use + operator, the magic method _add_ is automatically invoked in which the operation for + operator is defined.

Overloading binary + operator in Python:

When we use an operator on user-defined data types then automatically a special function or magic function associated with that operator is invoked. Changing the behavior of operator is as simple as changing the behavior of a method or function. You define methods in your class and operators work according to that behavior defined in methods. When we use + operator, the magic method _add_ is automatically invoked in which the operation for + operator is defined. Thereby changing this magic method's code, we can give extra meaning to the + operator.

How Does the Operator Overloading Actually work?

Whenever you change the behavior of the existing operator through operator overloading, you have to redefine the special function that is invoked automatically when the operator is used with the objects.Method Overloading:

Two or more methods have the same name but different numbers of parameters or different types of parameters, or both. These methods are called overloaded methods and this is called method overloading.

Like other languages (for example, method overloading in C++) do, python does not support method Overloading by default. But there are different ways to achieve method overloading in Python.

The problem with method overloading in Python is that we may overload the methods but can only use the latest defined method.

14.2 Method Overloading:

Two or more methods have the same name but different numbers of parameters or different types of parameters, or both. These methods are called overloaded methods and this is called method overloading.

Like other languages (for example, method overloading in C++) do, python does not support method overloading by default. But there are different ways to achieve method overloading in Python.

The problem with method overloading in Python is that we may overload the methods but can only use the latest defined method.

Object orientation:

Object-oriented programming (OOP) is a method of structuring a program by bundling related properties and behaviors into individual objects. In this tutorial, you'll learn the basics of object-oriented programming in Python.

Conceptually, objects are like the components of a system. Think of a program as a factory assembly line of sorts. At each step of the assembly line, a system component processes some material, ultimately transforming raw material into a finished product.

Inheritance:

One of the core concepts in object-oriented programming (OOP) languages is inheritance. It is a mechanism that allows you to create a hierarchy of classes that share a set of properties and methods by deriving a class from another class. Inheritance is the capability of one class to derive or inherit the properties from another class.

Benefits of inheritance are:

Inheritance allows you to inherit the properties of a class, i.e., base class to another, i.e., derived class. The benefits of Inheritance in Python are as follows:

It represents real-world relationships well.

It provides the reusability of a code. We don't have to write the same code again and again. Also, it allows us to add more features to a class without modifying it.

It is transitive in nature, which means that if class B inherits from another class A, then all the subclasses of B would automatically inherit from class A.

Inheritance offers a simple, understandable model structure.

Less development and maintenance expenses result from an inheritance.

Python Inheritance Syntax

The syntax of simple inheritance in Python is as follows:

Class BaseClass:

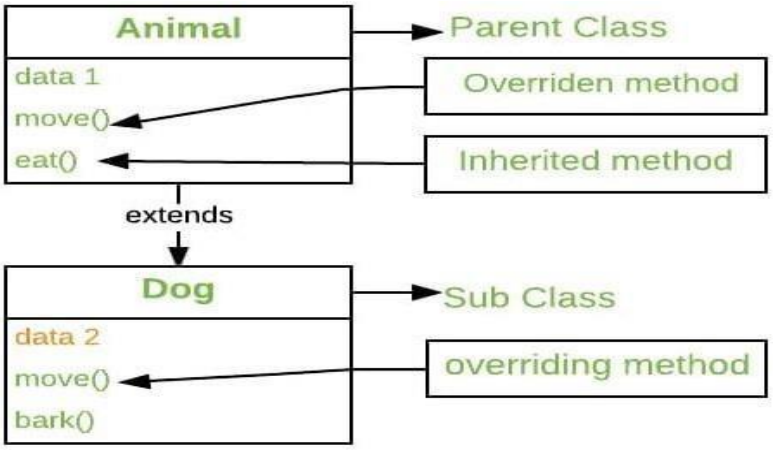
{Body}

Class DerivedClass(BaseClass):

{Body}

14.5 Method Overriding:

Method overriding is an ability of any object-oriented programming language that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes. When a method in a subclass has the same name, same parameters or signature and same return type(or sub-type) as a method in its super-class, then the method in the subclass is said to override the method in the super-class.



The version of a method that is executed will be determined by the object that is used to invoke it. If an object of a parent class is used to invoke the method, then the version in the parent class will be executed, but if an object of the subclass is used to invoke the method, then the version in the child class will be executed. In other words, it is the type of the object being referred to (not the type of the reference variable) that determines which version of an overridden method will be executed.

WEEKLY REPORT

WEEK – 15 (From Dt…04/04/24 to Dt 15/04/24….)

15.1 Abstraction

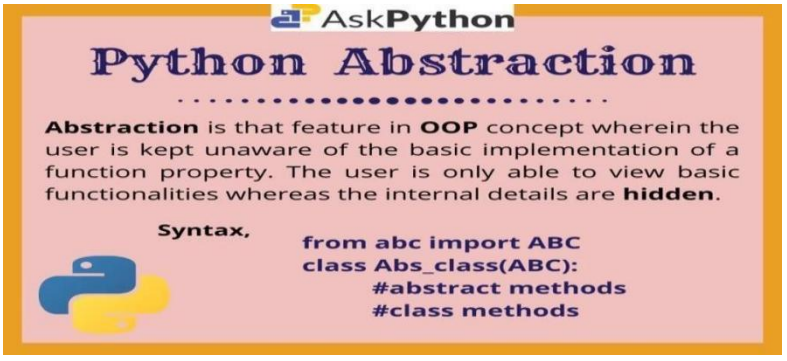
Data abstraction is one of the most essential concepts of Python OOPs which is used to hide irrelevant details from the user and show the details that are relevant to the users. For example, the readers of geeksforgeeks only know that a writer can write an article on geeksforgeeks, and when it gets published readers can read the articles but the reader is not aware of the background process of publishing the article.

A simple example of this can be a car. A car has an accelerator, clutch, and break and we all know that pressing an accelerator will increase the speed of the car and applying the brake can stop the car but we don't the internal mechanism of the car and how these functionalities can work this detail hiding is known as data abstraction.

To understand data abstraction we should be aware of the below basic concepts:

OOP concepts in Python

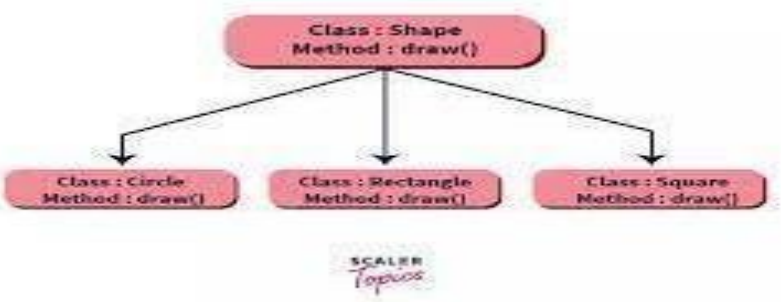
Classes in Python



Abstract classes in Python

15.2 Polymorphisom:

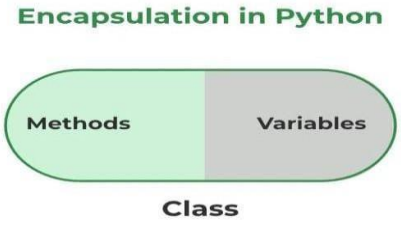
The word polymorphism means having many forms. In programming, polymorphism means the same function name (but different signatures) being used for different types. The key difference is the data types and number of arguments used in function.



15.3 Encapsulation:

Encapsulation is one of the fundamental concepts in object-oriented programming (OOP). It describes the idea of wrapping data and the methods that work on data within one unit. This puts restrictions on accessing variables and methods directly and can prevent the accidental modification of data. To prevent accidental change, an object's variable can only be changed by an object's method. Those types of variables are known as private variables.

A class is an example of encapsulation as it encapsulates all the data that is member functions, variables, etc. The goal of information hiding is to ensure that an object's state is always valid by controlling access to attributes that are hidden from the outside world.



15.4 Exception:

Exceptions: Exceptions are raised when the program is syntactically correct, but the code results in an error. This error does not stop the execution of the program, however, it changes the normal flow of the program.

Example:

Here in this code a s we are dividing the 'marks' by zero so a error will occur known as 'ZeroDivisionError'

```
marks = 10000
a = marks / 0
print(a)
```

Output:

```
Traceback (most recent call last):
  File "/home/f3ad05420ab851d4bd106ffb04229907.py", line 4, in <module>
    a=marks/0
ZeroDivisionError: division by zero
```

Types of Exceptions: