

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

ANFERA

(1)

COURSE PYTHON

PROGRAMMING

LECTURES (1 TO 7) 3 Weeks Content

INTRODUCTION

- ⑥ Is a Language To give a "instruction" of "Computer" To Perform a Specific Task.

Questions:-

- ① Why are Learn Python Programming ?
 - It is very Popular versatile language now.
 - very easy to understand like English like words. natural language

(2)

→ The Support of Community
is Top of Chat.

Application:-

Commonly used in App development, web development, Data Science, AI, Machine learning. Multiples doors when you learn for careers.

Whom Peoples don't Take this Courses?

→ None Programmers: Those who are passionate about creative arts but have no

Non-Technical Careers.

learning technical programming concepts and logical thinking.

→ Management Roles, Executive Roles not design this courses. who don't need programming skills in their careers.

Who is This Course for?

→ Development → AI Engineers
→ Software Engineer → Data Engineers
→ Students / Professional

Benefits of this Courses

Different fields like domains. Machine Learning, Deep Learning, Data Engineers, NLP they worked in all above fields. BackEnd, Process Automation

How to (SET-UP) Python In your Computer?

→ We have Study diff software installed in our PC. Using diff aspects.

Step 1:-

- Install Anaconda / Miniconda
- Download (IED) and install of Visual Studio
- After Then first Programms RUN in Computer. After that you can checked in your PC To install python

What is Anaconda

- Anaconda is a comprehensive Python distribution for data Science and Programming

→ Python of different packages anaconda can helping it without installation Python management, virtual environments very to easy.

2 Option in your Systems:-

- Can install "anaconda" or "Miniconda"

→ Anaconda basically whole packages of Python libraries around 1GB. Different softwares like Jupiter Note, Spyder support while → Miniconda is a smaller version of just "programming" doing with irrelevant packages. (unnecessary) minimum space taking easily install

To Checking in your Computer

(4)

Python is install successfully.

- Click on Start Button and then Searching Anaconda Prompt. And Open it.
- Write a Commands Python (-V) or python --version

Step 2:-

- Download IDE and install.
Basically means Integrated Development Environments. All Tools, Software are Required during longer. are Present there (Supports).

Different Tools:- Benefits

- Coding writing, debugging and execution in one place. VS code PyCharm. UI/UX, Multiple files project.

VS Code:- It is very

easy to use and "versatile". over worldwide. for

Python Programming. Extensions Install provided and customization.

VS Code Provide good manageable.

- Using diff language to follow Syntax otherwise code not run in PC. → Show that all guideline you is follow it.

"Syntax grammatic" miss out

of language it indicate this line of code is not correct.

Creating a folder and python

file in VS Code:-

- Open "VS code" → New file
- Create folder → Name file

(5)

Name your file using extension
•py like "hello-world.py"

Note:-

- VS Code Support Multiple languages, you install implicitly python Extension. (Left Side Panel) extension option there.

~~~~~

VIRTUAL Environment:-

- When we doing Programming we are facing Major Issues. Working one project in (1 Programming) and similarly doing (other Project). Python provide different libraries in different version.

Working Project A and Libraries install Numpy = 2.2 version. Similarly Project B and install Libraries install numpy = 1.3. System RUN both function Simultaneously. not RUN at a time not working on Project because Conflict.

So we have make a virtual environment from Project 'A' and 'B' not effect to each others.

Using Anaconda Prompt

Open Command line interface (CLI). black Screen.

Tick:- CL mai direkti likhna kya bad python (Press Tab) automatically next work - programming (or, programming)

(6)

Benefits of Virtual Environment:-

- ① Different Libraries Install using different version of Python in one virtual Environment.
It means different interpreter using different own Python interpreter.

`env = Name of virtual Environment`

Question: how can we checked your Active virtual environments?

Left side pr (base) environment it is whole system environment by default install in your Systems.

- ② when you install diff Libraries in Base environment System show conflict. name your envir.

③ Package install `run=22.2`
Using Pip, Conda. Command in Python. Pip is stable one and minimum conflict.

- ④ Pip install, conda install
- ⑤ Checking install using command `Conda pip list`

- ⑥ More then one environment
`Conda info -e`
- ⑦ Delete environment
`Conda remove --name (...) --all`

Variables:-

- ① A Containers which can store the values temporary of date. information

(7)

Variables dynamic nature
during Program the values
of variable change in executions.

Techniques:-

Standard methods used

Snake-Case:-

Ex: (user-name) word separated
by underscore. (lower case)

Camel-Case:-

Ex: (userName) first words, lowercase
others capitalized with no spacing

Pascal Case:-

Ex: (UserName) all words Capitalized.
with any spacing, underscore.

→ all cases using Naming in Variable

Follow one Conventional method
in code like Pascal Case.
Using throughout the code
This is Best practice.

Naming Rules:-

- Start with Letters or Underscore(-) (name, -name)
- No numbers at the Start. in variables names. (1name)
- Names can only contains letters numbers, underscore. (name1, user-name)
not use (User-name, user\$name)
- Case-Sensitive Variable are Case-sensitive
(Name, name) are diff Create
Upper Case and Lower Case

IMP Points:- Any language.

(8)

- Avoid Reserved Keywords like print, if, else, def by default words developer Sides commands. (def=10)
- Readable and Meaningful name of variables of code Readability: (total-price)

How To Create Variables:-

③ There are 3 things required.

→ Name → Assign operator (=) → Value

Ex "fruit = Apples"
 a = 10

Data Types:-

→ Integer Data Types:- Wholes numbers in Python. Used for counting

indexing and mathematical operations

Ex:- (-1, 5, 200) up To so on.

→ Float Data Types:- It represent "decimal" number (points values) used for more precise calculation. (3.14, -0.02, 3.5)

String Data Types:-

③ Sequence of characters enclosed in Single, Double quotes (''), ("") , "123" "Pakistan" as a Text

Boolean Data Types:-

③ Two values (True, False) condition checking.

Python NoteBook Execution:-

(.ipynb) Basic learning Start to execute a repeatedly Task so we use notebook.

How To check your variable type in programm.

```
Print("num1:", num1, typ(num1))
Message
Display my side, name , function()
```

Best Practices:-

→ (f-string) formatted Strings for Readability introduced by (Python 3.6+) version.

→ Customs Message display "age" and "change" and repeat value everytime.

→ Print(f"my age is {age}")

For Boolean :-

→ Use descriptive names: instead of flag or status, use is_active or has_errors

→ Avoid Comparing Booleans to True & False directly.

Keep Booleans purely boolean:

→ Resist the temptation to mix them with numbers. Although True behave like 1 and False is 0, it makes the code less clear.

→ None-type: absence of value
use: often used as a "placeholder" for "Optional" or "missing data".

→ checking for None with "is None" rather than == None

→ If you get a NoneType error
- It's None just.

Collection Overview:-

→ While Primitive data types we study (like int, Bool, float, str) hold single values.

(10)

① Python Built in collections let you group multiple items in flexible ways that's '4' Built in Collection is as:-

List	An order, mutable, Sequence
Tuple	An order, immutable Sequence
Dictionary	An unorder set of key-value pairs
Set	An unorder collection of unique elements

*List:-

- A list is created by [] Square brackets or the list() constructor.
- Ordered, meaning elements retain the orders in which they're creation inserted.

→ It is "mutable" you can add, remove or modify elements after creation.

Notes:-

② If we use (-ve) numbers use with index which means (Revers elements) pick.

→ append function:-

One existing list at the End add any values

→ (Insert value)- add '25' into index 2.

→ (Remove (20))- Remove the first occurrence of 20.

→ Pop()- Remove and Returns the last item.

Modifying :- Change elements at index (1 to 25)
numbers 1-11=25

* Tuple :-

- It is created using Parentheses () or the tuple() constructor.
- It is order similar to list.
- It is immutable (unchanged) like can't (add ; remove) once you defined creation & basic operations.

* Single Tuple:-

- Single-tuple element create using Parenthese with Comma like (2,)

* Slicing Concepts:-

- It's create (Sub-tuple) using index in Parenthese with Commas
Sub-Tuple = number_tuple[1:]

* Dictionary :-

(11)

- It is created using curly braces {} with (key-value) pairs. or using dict().
- It is unorded (Python 3.7+ give support in sorted) Keys must be unique and immutable (e.g strings, numbers, Tuples) value can be anything.
- Mutable can be changed add remove key pairs.

* Set:-

- A Set is created using curly braces {} or the set() empty set
- It is unorded and no allow duplicate elements.
It's mutable, you can add, change, remove

* Set Operations:-

- It supports mathematical operation like, Union intersection difference etc.
- Union :- all elements from both sides (Set-a | Set-b)
- intersection :- Common elements to both sets. (Set-a & Set-b)
- Difference :- Elements of Set-a but not in Set-b (Set-a - Set-b)
- Symmetric Difference :- elements in either Set-a ~~or~~ Set-b, but not in both. (Set-a ^ Set-b)
- Type Conversion (casting) :-
you can convert data types using built-in function like int(), float() and str(). etc

Operators

There are supports different operators like Addition, Subtraction, Multiplication, Division, Comparison operator, Logical operators. The purpose of these operators in Python diff data/variables performs calculation and logicals build in enhance way.

Assignment Operators:-

Are used to (=) Assign values to variables. For Exp.

Compound Assignment :- $x = 5$

$+ =$ adds and assigns $x + = 3$ equivalent in one step. to $x = x + 3$

Arithmetic Operators:-

- Are used mathematical operations like $+, -, *, /$
models, $\%$, floor division $//$ →
[without floating values answer give]
exponentiation $**$ (Double struck)

Logical Operators:-

- It is used for logic building when we used "more than one condition" to "compare it or mixed" it. $(\&, \text{or}, \text{not})$ are combined X Condition.

Comparison Operators:-

They give us Boolean output either True, False
using these operators in Python certain logical building's
use for test condition

- greater than ($>$), less than ($<$), ($=$)
- exclamation ($!=$) [not equal to]
- ($>=$) greater than and equal for case
- ($<=$) less than and equal for now
- ($=$) Two value comparison [equal or not equal]

Loops:-

- Repeated Execution :- Loops are enable Repeated execution of code blocks.
- Use Case:- Processing and automatic Task.
- Benefits:- Enhance efficiency, Simplicity, flexibility in programming

For-Loop:-

- It is for iterating over a **Sequence** (Lists, Strings, Tuples, dictionary etc.). It processes each elements one at a Time in very readable format.

Syntax :-

```
for items in Sequence
    # Process items
```

Indentation is by default Cursor appears after indentation. Show that every line of code execute after indentation [] execution in For Body.

Enumerate() in Loops:-

- You need both its **item** and **index**.

- This built in function make easy. in any collection.

Syntax :-

```
for index, items in enumerate(item):
    index System start [0, 1, 2, ...]
```

While Loops:-

- Repeats execution while a **Exp.** Condition is **True**.
- Used it when the numbers of iteration is unknown.

Syntax:-

```
while Condition
```

code block

→ Loops "Control Statements"

Used in above case (**infinite loop**)

④ Control Loop

Break Statements :-

- Break: Exits loops immediately.
 - Continue: Skip current iteration.
 - Else: Executes if loop completes normally.
-

Lecture NO # 8+9

Condition

Statement :-

- Are used to make decision in a program based on specific condition. It allow program to specific execute

different blocks of code depending on condition (True or False).

* Real-Life - Example:-

- If it rain, Take an umbrella
- If the traffic light is "green", drive, if it "red" stop. Green To Run.
- if the user enter the correct "password", allow "login" otherwise deny access.

* Basic Condition Statement:-

- Main three condition Statement.
- if → else → elif

Key insight:- It change the sequence of program execution based on condition.

How if-elif-else Evaluate Condition

- Python's if-elif-else structure works by evaluating conditions one by one in sequence. Each condition must contain ultimately resolve to either ('True or False)'

:- If "condition" is True. Execute the block and "skip the rest"

:- if all condition is False. The else Block executes (if present)

:- Only 'one' condition is executed in an (if-elif-else)

Case Sensitive :-

You words/characters is lower case or upper case.

Nested Conditions:-

- One if condition written can multiple if condition
[Condition one or multiple condition]

Functions:-

- A function is reusable block of code designed to perform a specific task.
 - Same input and same output
- It break down a large problem into a smaller manageable parts
- They make code easier to read, debug and maintain
- When a function is called, it runs a "set of instruction" and may return a result.

Benefits :-

- Code Read/Reusability
- Concise, clean and Modular
- Error Reduction.

Work :-

- ① You defin it with a name and it take inputs (parameters) and produce output. (change other condition.)

Example:-

- Input : A fruit
- Output : Made a juice in a Juicer machine. To performs function next Time you can Reuse it.

Types :-

- (1) Built in function
- (2) User-defined

Built-in-function

- Already Predefined function in python (ready made recipes)

Exp:lower() function

- Print() : Every string to display.
- input() : User Take input.
- type() : To find type of variable
- length() : To check in variable/list/tup
- Max() : Max number Show
- Sorted() : Lowest to highest num
- Range() : Specific to generate a range
- id() : Return memory address of an object
- eval() : Any math expression in String form To pass eval()

User-defined-fun

- function created by the programmer
- Custom recipes

Lectures No# 9+10

Topic:-

User Define

Functions

In this functions, User can defined User defined function which means

User can do having like
First Set of Rules:-

- (i) → Purpose of function what can do.
- (ii) → Function Name - Use a useful name for example
 $\text{Sum} = a + b$, $\text{add} = a + b$

'()' Parentheses after names meaningful name is best practices.

- (iii) → Input Parameters of function To defined it To Take input inside it.

Defin Parameters:-

Take Function as a input is called Input Parameters of function.

- (iv) Apply logic in function
- (v) what shown in output (Return is decision on this stage)

A function is a block of reusable code that performs Task.

Best Practices :-

Note: Write in your Task in Purpose
Multiline string to (" ")
memory you works In comments

→ calling by Name of that
function after defin.

→ 1st block = function defined

2nd block of code = call function

→ Return statement is optional

Statement print : Function

جیسا کوی اسکرپٹ میں بھی اسکے

بھی اسکے لئے print function

کوی اسکرپٹ میں اسکے لئے statement

call کوی اسکرپٹ میں just value

کو Statement (print) کو اسکے لئے

Define function with -

Max element
Min element

(29)

Default Argument / Parameters:-

- By default values function has passing it

Keyword Arguments:-

- AT the Run Time any input pass it.

Required Arguments:-

- if required Argument is not pass during the call of function it Refused. during execution of code

Variable length of Arguments:-

- we have defined length in parameter of variable then function performed it.

Global Vs Local Variable

(1) IMPORTANT LOCATION

- Defined outside function and can be accessed anywhere
- Defined inside a function and cannot be accessed outside.

Example:-

A "Teacher" Something written on white board and his student read/write lectures on a specific time and not all the students of A School read not. "Board class the outside not visible" is called Local variable "Notice Board" on School. any information given by administration. Every one can access called Global variable. Changed information only administrations.

- If Classrooms is function if one ^{inside} variable then access only this function element can changed.
- If outside of function is consider Global variable. The value can changed inside it role is different.

(2) HOW CAN ACCESS THE VALUE AND CHANGED IT

if variable outside function	if variable inside function	if variable in Nested variable carry with 2-variable
global variable	is local variable	

Key Difference

- "is": check if two object occupy the same memory location
- "==" : check if two object have the same values

Lecture No#11+12

Topic:- What Are Higher-Order Functions

• Function that take another function as an argument or return a function as output.

* Common higher Order functions:

→ `map()`: Apply a function To every item in an iterable

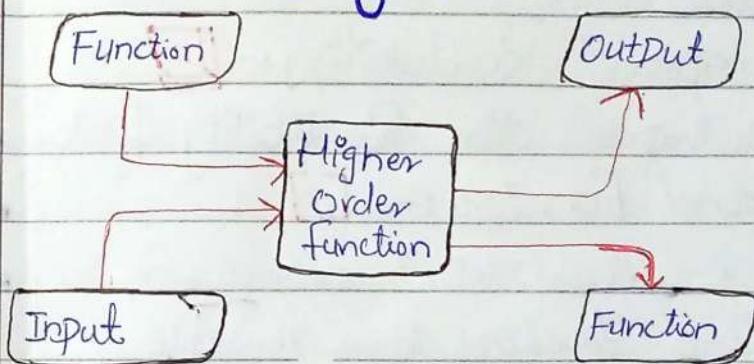
→ `filter()`: Filters elements based on a condition

→ `reduce()`: Reduce a Sequence To a Single value

, `zip()`: Combines multiple iterables into Tuples. (Pairs bracketing)

, `enumerate()`: Add an index To an iterable.

Diagram



In Simple words.

① instead Taking the value of taking as input or give output as function

Example:-

② One organization and head of department in every one Task To Take head of department and given Task To given Teams members. The output will be call `higher-order function`

What are Decorators?

★ Modify Behavior:- 🔑

- Are those functions that modify the behavior of other functions without changing their code.

★ Improve Readability:- 📄

- It improves the readability of code.

★ Extend Function :- []

- They allow us to extend or enhance functions in a reusable way.

★ Key Function / Features:- ↗

- Promote Code reusability.

★ Common Uses:- ⚙️

- Used for Logging, authentication, timing function and more.

★ Why Use To Decorators

→ Avoid Code Duplication

- Add extra functionality without modifying the existing.

Use Cases:-

→ Logging function execution

→ Measuring execution time

→ Checking User authentication

→ Before executing a function

Example:-

ابن فتحی کو بایکو لوگو رکی برائیں الفریض

کوئی نہ اور بینی بھی میر خیس فیکٹری میں اس نے

دوسری تین میں میں (Age, Gender, Name)

نام، موبائل نمبر اور تسلیم کیسے (Name, Phone No, Gender)

نام، موبائل نمبر اور دستیاری (Name, Phone No, Designation)

کے ساتھ کچھ اضافی میں کو اڈ کر کر (Based on CS) (Add on CS)

gender میں ایک سے جو ایک gender
same different میں ایک سے جو ایک gender
(function+Decorators) اس کو decorator کہا جاتا ہے

Syntax:-

Syntax: - پہلے خانش ساخت میں جس کو Behavior و نسل کرنے کے دوبارہ دیکھنا۔ اسی میں دوسرے دلے دیکھنے کا wrapper فونکشن کو support کرنے میں کام کرے گا۔

④ Logging IS that:-

جس میں ہم سُمیٹ رکھیں اور کوئی Error رکھ رہے
ہوئے ہیں فائل کو جدید ریکارڈ پوتے تو اس میں جتنی بھی Major
ہیزر میں پوتی ہے۔ اسے سُمیٹ میں وہاں کروالتے ہیں۔
Rکہ یہ سے اس نام کوں ساونٹن رہا ہے۔ اس
سے Output جیکہ اس کی کی Functionalities
کوئی اس پر کوئی مشکل نہیں۔ اس کو Run (Run) Successfully
کہا جائے۔ اس کو جیکہ کرنے والے نہیں تو یہ جیکہ کرتے ہیں کس
User نے کس وقت کوں اسے functionality کو جیکہ کیا تھا۔
اس ہیزر کو دیکھنا آسان یہو جائے ہے۔

④ Multiple Decorators :-

⑥ You can use multiple decorators in function stack to each others from bottom to top. The decorators closest to the function is applied first, one by one.

Real World Examples :-

- (i) Logging :-
- (ii) Pre-processing :-
Recording patient details
- (iii) Main Task :- Consulting the patient
- (iv) Post-processing :- Writing prescription
- (v) Timing & Consultation Time

(Main Function क्या एवं :-)

→ Similarly one function can do many Task Additional Task like

- (i) How Much Time (ii) Run Time Input
- (iii) Function Logging from User. up to soon

Meta Data of function :

- A Data with Associated Data is called Meta Data.

Real-World-Examples :-

(i) Logging :- (ii) Pre-processing :-

- Recording patient details
- Sanitize hands

(iii) Main Task :- (iv) Post-processing :-

- Consulting the patient
- Writing prescription

v) Timing :- Consulting Time

(Main Function Ky ekha :-)

→ Similarly one function can do
many Task Additional Task like

(i) How Much Time (ii) Run Time Input

(iii) Function Logging from User up to ~~5000~~ 5000

vi) Meta Data of function :

A Data with Associated Data

is called Meta Data.

Lectures No #13+14

Real-World Examples:-

- Logging :- (i) Pre-processing :-
Recording Patient details | Sanitize hands
- Main Task :- (ii) Post-processing :-
Consulting the patient | Writing prescription
- Timing :- Consulting Time
(Main Function क्या एवं :-)
→ Similarly one function can do many Task Additional Task like
 - (i) How Much Time (ii) Run Time Input
 - (iii) Function Logging from User. up to soon
- Meta Data of function :

- A Data with Associated Data is called Meta Data.

Topic:- File (Input/Output) In Python

Topics Covered :-

- What is File I/O? Why do we need file handling?
- Opening and closing files in Python
- Working with "JSON" files in Python
- Best Practices in file handling
- Exception Handling
- Practice Exercise

File I/O:-

Definition: It allows a program to interact with files of external source.

Input :-

Reading from a
file in an example
of input.

Python provides built-in function like `open()` for file operation.

Why do We Need it?

D^(safe)eristant Storage

To store and
retrieve persistent
data

→ Large Data Set

Saves User Preference
in Application

Output :-

Writing to a file
is an example of
output

Common Examples
Reading Text, writing
data, handling
JSON/CSV files.

Data Sharing

Enables Sharing
data b/w different
Programs.

User Preference

Useful when working
very large data

Logging :-

- Stores logs for future analysis or debugging.

● بعض اوقات یہیں Data کو سی جگہ پر Safe کرنا
جائز نہیں بلکہ external storage کے ذریعہ سے (file)
جائز نہیں جو کہ دیکھا کو سلوک کرتے ہیں وہ عارضی ہو جائیں
جس بروگرام کی اینٹریشن ختم ہو جاتی ہے مگر well-formed value
جانب پر اس کے بعد یہیں ایسے طریقے کی ضرورت نہیں ہیں
جائز کو کروایا سکتے۔

→ Solving Problems Using database

Storage for Permanent Storage.

Logging حسٹھ کوئی ایکسچینج جلتی ہے تو اس کی ریکارڈیng

وہ کمرتی سیر - اس Time پورے کونا سے Action کو لے لے

لے اس میں Error 0 باعث ہے جو Server کی وجہ سے ہے

بی سوئی ہے جس نویں بعد میں Application میں جو اس کو رکھا

اس کو جس کر کے - اس کو ^{internal} خالی کے ساتھ interact کرتا

elijah modify elijah safe

→ Working With JSON files:-

JSON (Java Script Object Notation)

It is used to exchange and store in popular way. export any language in working and developed any APIs and exchange data from one platforms To another effective and useful formats.

Same as dictionary like Key-values pairs performed.

* Exception Handling

Allows programs To handle errors gracefully instead of crashing. It ensure the program continue running even if an error occurs.

uses keywords like Try, except, finally.

→ Example-Real Life- ATM

- (1) System should not crash on invalid input
- (2) If you enter -ve amount you get a msgs not a crash
- (3) Shows a helpful msgs like Invalid Amount.
- (4) Helps in building user-friendly software
- (5) Prevent Program termination in such scenarios.

→ Types of Errors:-

Syntax Error (Parsing)	Runtime Errors (Exception)
----------------------------	--------------------------------

Lectures No # 15 + 16

Topic:-

Object Oriented Programming (OOP)

Questions:-

Why Need To Read it?

If We not Use it What happens? Which Types of Issues are facing?

آپنے ایسے سورج Tamzin

For example:-

- Profile ایک سٹوڈنٹ کا دلیل

McGrend Name اس کا نام Save اس کو ذخیرہ کرنا

- بعد میں اس کو استعمال کریں - یعنی Variable

Students 1000 اب - Now

کے بینوں Data کو سرسری کرنے کا کام کیا جائے۔ ویری ایبل نہ
Copy, paste اس کا شروع (Copy, pasting) کر کر fill کو Data
MongoDB کو اس کے مقابلہ میں لے لے جائے۔ اس کو (MongoDB)
کرنا (IMPOSSIBLE) سمجھ کر بوجانہ کا۔ سطح پر جو کوئی بھی
یہ بھگہ بنائے بشرت۔ اس کے لئے بھرپور Real Life Examples
کرنے کا handle اس problem scenario

Real life کی تینی کامیابیوں کی دلیل

Support کر کر Solve کی Situation - کمی میں

OOP کا functionality کی دلیل

Some Terminology In OOP :-

we have studied in OOPS Terms

→ **Classes:-**

→ **Attributes** are the data stored inside an object.

→ **Methods** define the behavior of object.

→ **Objects** is a specific instance of a class.

Class:-

- Imagine you have build multiple houses in this area.
- A Block has same structure houses and internally customization individually.
- 1st Make Blue print (نمونہ) house structure on paper build by architectures
- We have used Blue print To build multiple house with same structure. Blue Print is (Class) (Schema functionality) used To drive next functionality.
- Class Asi Jeezein Jis ko hum define krtay. (Blue print) design krtay.

Attributes:-

Example:-

Car Manufacturing ③
 ۱. Car Company ۱
 ۲. Blue Print ۶ (Company)
 Same ۷ (Based on Class ۸)
 Car ۹ (Cars)
 اس کی خاصیت ۱۰ (Attributes)
 Design ۱۱ (Structure, colors)
 اگر کسی کو دیتا جائے

Methods:-

اوپر والی کی فناوری پر اس ۱۲ (Perform وغیرہ) functions کی
 unlock, lock ۱۳ (Car)
 - (Method) ۱۴ (Real Life)
 ۱۵ اسکلر ۱۶ (Real Life)
 ۱۷ Student ۱۸ (Design of Management)
 ۱۹ Student Marks, gender, attributes, Name, Etc.
 - ۲۰ Attributess: ۲۱ Individual

Methods

Methods Allow us to do Some
Attendance of student is same
(Want class) for pass Exam -
Cross all students (Rail and Regulation
attribute of Cars follows
function of class
(OOPS) Language handle Large
scale solutions to solve many kind

- Code ki Readibility and Usability is more better.
- Manageable with better way To Right. Create (classes) and Tell us the (attributes) of that classes.
- Define (Methods) with function is performed in (Code To Reuse).

for different Students.
Not (Copy Paste) our code
To handle (Simple and hard Solution) To Solve it.

→ Class: Cars

→ Attributes: Color, height, weight, fuel, Engine: capacity and Speed. defined it.

→ Methods: To defined cars AC, car

Mirrors reactivable etc. strings

→ Attributes: size (size is defined method)

★ How To Use it :-

→ Object Oriented Programming

Object Create in every class. ①

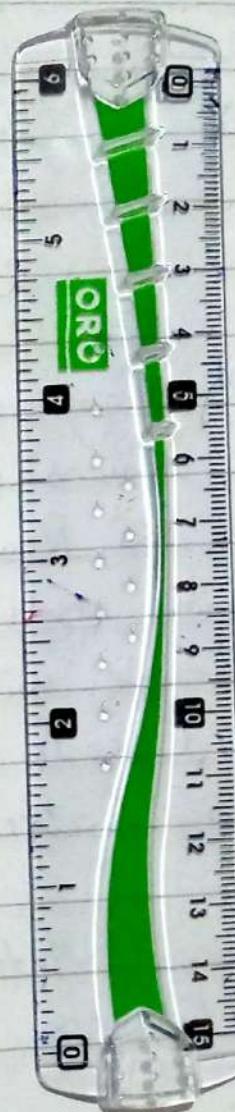
Using (class) in blue print? Actual object in Actual we use (inference, object) for

طبع (Blueprint) ای
Object کو (object) کے Class کی
بیوئی (entity) کا (inference Object) ہے۔
1. جسے کہا جائے

Previous Lectures we have
studied Core Objectives of OOP
have discuss with Real Life Example
Now we have studied is.

A What Are we doing :-

- Implementing OOP in Python.
 - Practicing with classes and objects
 - Using __init__ Constructor
 - Exploring object Method.
- Practices on Python
Notebook in VS.



Person's Attribute: *private*