OPPS

Abstraction:

It is simply finding which functionalities should be hidden from the user and which should be shown to the user. It is a design level concept.

Access specifiers in python:

Using these specifiers we can make attributes and methods of class Protected and Private.
PUBLIC: The attributes and methods of a class can be accessed by anyone.
PROTECTED: These can be accessed by only the parent class and its derived class.
PRIVATE: These attributes and methods can only be called within the parent class.
We can't call the private attributes and methods outside the class means we cant call with its own object also.
Syntax:
Protected: just use underscore in front of attribute or method name. Like _name or _display()
Private: __name or __display().
There is No strict rule in PYTHON that we can't access these protected and private methods or attributes.
Now the question comes "what is Point in creating Protected and private"?
ANS: It is the responsibility of a good programmer to not touch those methods or attributes.
How to access private or Protected?
Use 'dir(s1)' > Then it will show how to access.

Encapsulation:

What is a capsule?

A capsule is a combination or bundling of small components of medicine right! Which is called encapsulation.It means we are protecting the medicine. IT means medicine has security.

DEF: Rapping up of data or methods into a single unit is called encapsulation or it is also known as Data Hiding. It is an implementation level concept. Abstraction can be achieved using encapsulation

Following are the ways to access private data:

1. Creating a public method within the class and in it calling the private method (self.private_method()).
2. **Namemangling**: object-name._classname__privatemethodname() or attribute name
3. **Using Getters and Setters methods:**
   Getters: It is used to access the private information
   Setters: It is used to modify or change the information

```
def get_age(self):
    return self.__age
def set_age(self,age):
    if age>35:
        print("Invalid age given..Age should be less than 35.")
    else:
        self.__age=age
```

Inside the setters in the above code validation logic is applied.

- Encapsulation implements Abstraction. Abstraction is just a thought process. How to achieve encapsulation: obviously using the access specifiers

Advantages of Encapsulation?

- It provides security.
- It helps in data hiding.
- Remaining we get on practise

Polymorphism:

Poly→ Many
Morphism→ forms
Ex: Humans we behave differently under different situations like with friends, family, at the office etc. Means Humans are polymorphic in nature.

There are four ways to implement polymorphism in python:

1. Duck Typing.
2. Method Overloading.
3. Operator Overloading.
4. Method Overriding.

INTRO:

1. addition(+) → It adds two numbers and it concatenates two characters, which means it is polymorphic.
2. Length(len()) → len(string) = it will calculate length of characters | len(dict1) = It will give no.of key value pairs | len(list) = It will have no.of items in list, which means it is polymorphic.

1. Duck Typing in Python:

Why is duck typing used?
It allows the data type to be dynamic! Or it supports dynamic typing.
Dynamic: It is nothing but we don't need to mention the data type at compile time. At run time it automatically knows it. PYTHON, remaining all like c, c++, java are static typing languages!

- In duck typing the name of the class doesn't matter, what matters is the methods and attributes you defined.
- The class of the objects can change as long as methods are defined on that object.

-
```
def display(obj):
    obj.swim()
    obj.speaks()
    print("Information displayed")
d=Duck()
dog=Dog()
p=Person()
display(d)
display(dog)
display(p)
```
-
```
Traceback (most recent call last):
  File "C:\Users\prade\PycharmProjects\Polymorphism_demo\duck_typing.py", line 23, in <module>
    display(p)
  File "C:\Users\prade\PycharmProjects\Polymorphism_demo\duck_typing.py", line 15, in display
    obj.swim()
AttributeError: 'Person' object has no attribute 'swim'
I am a duck and I can swim
Quack Quack
Information displayed
I am a dog and I can swim
woof woof
Information displayed
```
- In the above code we are getting error only in case of person as person doesn't have any method swim() but we are calling it so we are getting error.

2. Operator Overloading:

- If the operator is performing beyond its capability then we say the operator is overloading.
- Before knowing how to do overloading we need to know about ‘magic methods’ or ‘special methods’ or ‘Dunder Methods(Double underscore methods).

Dunder Methods:

- 1. Addition operator (+): Behind the scene there is a method. In python everything is a class like int, float, str ..etc all are classes.
- print(int.__add__(1,2)) == print(1+2) similarly string method is also there.

Advantages of Operator Overloading:

1. We can easily define operators according to their use, all the arithmetic operators are familiar to us.
2. It improves code readability.

Method overloading:

Defining multiple or many methods using the same name with different numbers of parameters or arguments in the same class.

- Python doesn't support method overloading.
- We can achieve it by adding default argument

Method overriding:

There must be two classes. So it is implemented in only inherited classes.Here method name is also same and no.of parameters or arguments are also same. Now how are these methods different?: They differ in location, one will present in parent class and other will present in child class.

- Method overloading is compile time polymorphism and Method overriding is a Runtime polymorphism.

File Handling in Python:

Ram: It is a volatile memory!
File: It is simply a container or name memory location where you want to store data or it is just a sequence of bytes where you will store permanently.

- Types of File Handling:
- 1. Text files (like .csv, .py) we open using text editors
- 2. Binary Files (like images, audios, zip files etc)

Basic Operations on file:

- OPEN
- READ
- WRITE
- CLOSE.

Syntax:
file_object = open("filename","mode")
Mode can be:

- r→ read
- w→write
- a→append(adding something at the end of line or at last in program)
- r+ → read and write
- w+ → read and write

- a+ → read and append