

Abstraction

- 1 Abstraction in programming is a fundamental concept that involves simplifying complex systems by breaking
- 2 them into smaller ,more manageable parts while hiding unnecessary details.It allows programmers to work
- 3 with high-level,generalized representations of objects and processes,rather than getting bogged down in the specifics
- 4 of their implementation.Abstraction makes code more readable,maintainable and reusable.

Features of Abstraction

- 1 Simplifies Complex Systems
- 2 Hides Implementation Details
- 3 Promotes Reusability
- 4 Facilitates Collaboration
- 5 Enhances Maintenance
- 6 Improves Code Understandability
- 7 Enforces a clear seperation oof Concerns
- 8 Adapts to Changing Requirements

```
In [3]: 1 from abc import ABC, abstractmethod
2
3 class info(ABC):
4
5     # abstract class: --->Implemented + non-implemented method
6
7     def show(self):
8         print("This is implemented method from info Abstract class")
9
10    @abstractmethod
11    def display(self):
12        pass
13
14    class read(info):
15        def display(self):
16            print("This is non-implemented method from Abstract class: implemented by read class")
```

```
In [4]: 1 #Cannot create abstract class object
2 i1 = info()
3
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[4], line 2
      1 #Cannot create abstract class object
----> 2 i1 = info()

TypeError: Can't instantiate abstract class info with abstract method display
```

```
In [5]: 1 r1 = read()
2 r1.show()
3 r1.display()
```

```
This is implemented method from info Abstract class
This is non-implemented method from Abstract class: implemented by read class
```



```
In [8]: 1 class RBI(ABC):
2
3     #interface :--->only have non-implemented method
4
5     @abstractmethod
6     def rateOfInterest(self):
7         pass
8
9     @abstractmethod
10    def withdraw(self):
11        pass
12
13    @abstractmethod
14    def deposit(self):
15        pass
16
17    class Axis(RBI):
18
19        def CustInfo(self):
20            print("Welcome to Axis Bank")
21
22        def rateOfInterest(self):
23            print("Axis Bank have Home Loan Rate on Int 8%")
24
25        def withdraw(self):
26            print("Axis bank have withdraw limit upto 50k per day")
27
28        def deposit(self):
29            print("Axis bank have deposit limit upto 90k per day")
30
31    class HDFC(RBI):
32
33        def CustInfo(self):
34            print("Welcome to HDFC Bank")
35
36        def rateOfInterest(self):
37            print("HDFC Bank have Home Loan Rate on Int 11%")
38
39        def withdraw(self):
40            print("HDFC bank have withdraw limit upto 40k per day")
41
42        def deposit(self):
```

```
43 print("HDFC bank have deposit limit upto 70k per day")
```

In [10]:

```
1 a1 = Axis()  
2  
3 a1.CustInfo()  
4 a1.deposit()  
5 a1.rateOfInterest()  
6 a1.withdraw()
```

Welcome to Axis Bank
Axis bank have deposit limit upto 90k per day
Axis Bank have Home Loan Rate on Int 8%
Axis bank have withdraw limit upto 50k per day

In [11]:

```
1 h1 = HDFC()  
2  
3 h1.CustInfo()  
4 h1.deposit()  
5 h1.rateOfInterest()  
6 h1.withdraw()
```

Welcome to HDFC Bank
HDFC bank have deposit limit upto 70k per day
HDFC Bank have Home Loan Rate on Int 11%
HDFC bank have withdraw limit upto 40k per day

In []:

```
1
```