

```
In [ ]: #Class Object, method, encapsulation, abstraction
```

Abstraction

Hiding unnecessary details(complex detail or code) and showing necessary details to the user.

Example:- CAR

Complex details or mechanism such as how engine is working, radiator is working etc is hidden from the car user(driver) in car body and only necessary details such as driving wheel, Accelerator,brakes, clutch, gear etc are given to the user. This is called as Abstraction

Example:- Mobile Phone

Print("msg",variable)

Definition of print() is hidden from the user(developer) and only necessary details i.e syntax of print() function is given to the user to use it.

```
In [1]: class character():
    def __init__(self,name):
        self.name=name
        self.__score=0
        self.__life=3

    def displaylife(self):
        return self.__life
    def displayscore(self):
        return self.__score
    def punch(self):
        self.__score=self.__score +5
    def kick(self):
        self.__score=self.__score +10
    def stabbed(self):
        self.__life=self.__life - 1

    def intro(self):
        print("Player Name:- ",self.name)
        print("Initial Score:- ",self.displayscore())
        print("Initial Life:- ",self.displaylife())

mario=character("Mario")
mario.intro()
```

```
Player Name:-  Mario
Initial Score:-  0
Initial Life:-  3
```

```
In [3]: mario.displaylife()
```

```
Out[3]: 3
```

```
In [4]: mario.intro()
```

```
Player Name:- Mario  
Initial Score:- 0  
Initial Life:- 3
```

```
In [6]: mario.kick()
```

```
In [7]: mario.displayscore()
```

```
Out[7]: 10
```

```
In [18]: class character():
    def __init__(self,name):
        self.name=name
        self.__score=0
        self.__life=3

    def displaylife(self):
        return self.__life
    def displayscore(self):
        return self.__score
    def punch(self):
        self.__score=self.__score +5
    def kick(self):
        self.__score=self.__score +10
    def stabbed(self):
        self.__life=self.__life - 1

    def intro(self):
        print("Player Name:- ",self.name)
        print("Initial Score:- ",self.displayscore())
        print("Initial Life:- ",self.displaylife())

mario=character("Mario")
mario.intro()
```

```
Player Name:-  Mario
Initial Score:-  0
Initial Life:-  3
```

```
In [19]: # x=mario.displaylife()

# if(x==0):
#     print("Game Over")
# else:
#     print("Welcome To level 2")
```

```
In [ ]:
```

