

Assignment-7(RID:001,Madhur Jodhwani)

1. Write a program which contains one class named as BookStore.  
BookStore class contains two instance variables as Name ,Author.  
That class contains one class variable as NoOfBooks which is initialise to 0.  
There is one instance methods of class as Display which displays name , Author and number of books.  
Initialise instance variable in init method by accepting the values from user as name and author. Inside init method increment value of NoOfBooks by one.  
After creating the class create the two objects of BookStore class as  
Obj1 = BookStore("Linux System Programming", "Robert Love")  
Obj1.Display() # Linux System Programming by Robert Love. No of books : 1  
Obj2 = BookStore("C Programming", "Dennis Ritchie")  
Obj2.Display() # C Programming by Dennis Ritchie. No of books : 2

Answer:

```
class BookStore:
    no=0
    def __init__(self,bookname,author):
        self.book=bookname
        self.name=author
        BookStore.no+=1

    def Display(self):
        print(self.book,end=" ")
        print(self.name,end=" ")
        print("No. of books:",self.no)

def main():
    obj1=BookStore("Linux System Programming","Robert J Love")
    obj1.Display()
    obj2=BookStore("C Programming","Dennis Ritchie")
    obj2.Display()

if __name__=="__main__":
    main()
```

Commented [MJ1]:

2. Write a program which contains one class named as BankAccount.  
BankAccount class contains two instance variables as Name & Amount.  
That class contains one class variable as ROI which is initialise to 10.5.  
Inside init method initialise all name and amount variables by accepting the values from user. There are Four instance methods inside class as Display(), Deposit(), Withdraw(), CalculateIntrest().  
Deposit() method will accept the amount from user and add that value in class instance variable Amount.  
Withdraw() method will accept amount to be withdrawn from user and subtract that amount from class instance variable Amount.  
CalculateIntrest() method calculate the interest based on Amount by considering rate of interest which is Class variable as ROI.  
And Display() method will display value of all the instance variables as Name and Amount.  
After designing the above class call all instance methods by creating multiple objects.

Answer:

```
class BankAccount:
```

```
ROI=10.5
```

```
def __init__(self, name, amount):
```

```
    self.Name=name
```

```
    self.Amount=amount
```

```
def Display(self):
```

```
    print(self.Name)
```

```
    print(self.Amount)
```

```
    print(self.ROI)
```

```
def Deposit(self):
```

```
    print("enter amount:")
```

```
    da=int(input())
```

```
    self.Amount+=da
```

```
def Withdraw(self):
```

```
    print("enter amount:")
```

```
    self.Amount=self.Amount-int(input())
```

```
def CalculateIntrest(self):
```

```
self.ROI=BankAccount.ROI*self.Amount
```

```
def main():
```

```
    obj1=BankAccount("Madhur",5000)
```

```
    obj2=BankAccount("Komal",7654)
```

```
    obj1.CalculateIntrest()
```

```
    obj2.CalculateIntrest()
```

```
    obj1.Deposit()
```

```
    obj2.Withdraw()
```

```
    obj1.Display()
```

```
    obj2.Display()
```

```
if __name__=="__main__":
```

```
    main()
```

Commented [MJ2]:

3. Write a program which contains one class named as Numbers.

Arithmetic class contains one instance variables as Value.

Inside init method initialise that instance variables to the value which is accepted from user. There are four instance methods inside class as ChkPrime(), ChkPerfect(), SumFactors(), Factors().

ChkPrime() method will returns true if number is prime otherwise return false.

ChkPerfect() method will returns true if number is perfect otherwise return false.

Factors() method will display all factors of instance variable.

SumFactors() method will return addition of all factors. Use this method in any another method as a helper method if required.

After designing the above class call all instance methods by creating objects of the class

```

class Arithmetic:
    def __init__(self,no):
        self.Value=no

    def ChkPrime(self):
        if self.Value > 1:
            for i in range(2, self.Value):
                if (self.Value % i) == 0:
                    print(self.Value, "is not a prime number")
                    break
            else:
                print(self.Value, "is not a prime number")

    def ChkPerfect(self):
        sum = 0
        for x in range(1, self.Value):
            if self.Value % x == 0:
                sum += x

        if(sum==self.Value):
            print(self.Value,end=" ")
            print("is a Perfect Number")
        else:
            print(self.Value,end=" ")
            print("is not a Perfect Number")

    def printFactors(self):
        i=1
        while(i<self.Value):
            if(self.Value%i==0):
                print(i,end=" ")
            i=i+1

```

```

        print()

    def FactorSum(self):
        sum = 0
        for x in range(1, self.Value):
            if self.Value % x == 0:
                sum += x
        print("Sum of Factors is:", sum)

def main():
    obj1=Arithmetic(65)
    obj2=Arithmetic(54)
    obj1.ChkPrime()
    obj1.ChkPerfect()
    obj1.printFactors()
    obj1.FactorSum()
    obj2.ChkPrime()
    obj2.ChkPerfect()
    obj2.printFactors()
    obj2.FactorSum()

if __name__ == "__main__":
    main()

```