*#Program*x=int(input("Enter the value to be checked as odd or even - "))  
def check():  
 if x & 1:  
 return 'odd'  
 else:  
 return 'even'  
  
*# print("Result is ",check())  
  
#Program*class Mystore:  
 \_\_prod\_code=[]  
 \_\_prod\_name=[]  
 \_\_prod\_price=[]  
 \_\_prod\_quant=[]  
  
 def getdata(self):  
 self.p=int(input("Enter no.of products you need to store: "))  
 for x in range(self.p):  
 self.\_\_prod\_code.append(int(input("Enter product code: ")))  
 self.\_\_prod\_name.append(input("Enter product name: "))  
 self.\_\_prod\_price.append(int(input("Enter cost price: ")))  
  
 def display(self):  
 print(" Stock in stores")  
 print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
 print("Product code\t Product name\tCost price")  
 print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
  
 for x in range(self.p):  
 print(self.\_\_prod\_code[x], "\t\t\t",self.\_\_prod\_name[x],"\t\t\t",self.\_\_prod\_price[x])  
 print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
 def print\_bill(self):  
 total\_price=0  
 for x in range(self.p):  
 print("Enter the Quantity of product code")  
 q=int(input(self.\_\_prod\_code[x] ))  
 self.\_\_prod\_quant.append(q)  
 total\_price=total\_price+self.\_\_prod\_price[x]\*self.\_\_prod\_quant[x]  
 print(" INVOICE RECEIPT ")  
 print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
 print("Product code\t\tProduct name\t\tCost price\t\tQuantity\t\tTotal Amount")  
 for x in range(self.p):  
 print(self.\_\_prod\_code[x],"\t\t\t\t",self.\_\_prod\_name[x],"\t\t\t\t",self.\_\_prod\_price[x],"\t\t\t\t", self.\_\_prod\_quant[x],"\t\t\t\t",self.\_\_prod\_quant[x]\*self.\_\_prod\_price[x])  
 print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
 print(" Total Amount=", total\_price)  
 Amount=int(input("Enter the amount given by consumer "))  
 Total= Amount -total\_price  
 print("Balance amount= ",Total)  
  
s=Mystore()  
s.getdata()  
s.display()  
s.print\_bill()  
  
  
*#Program -Create a Bus class that inherits from the Vehicle class. Give the capacity argument of Bus.seating\_capacity() a default value of 50.*class Vehicle:  
 def \_\_init\_\_(self, name, max\_speed, mileage):  
 self.name = name  
 self.max\_speed = max\_speed  
 self.mileage = mileage  
  
 def seating\_capacity(self, capacity):  
 return f" The Seating Capacity of a {self.name} is {capacity}"  
  
class bus(Vehicle):  
 def seating\_capacity(self,capacity=50):  
 return super().seating\_capacity(capacity=50)  
  
b = bus("Omni",150,250)  
print(b.seating\_capacity())  
  
*#program-Define a class attribute”color” with a default value white. I.e., Every Vehicle should be white.*class Vehicle:  
  
 def \_\_init\_\_(self, name, max\_speed, mileage):  
 self.name = name  
 self.max\_speed = max\_speed  
 self.mileage = mileage  
  
class Bus(Vehicle):  
 color = "white"  
  
class Car(Vehicle):  
 color = "white"  
  
c= Car("SWift ",170,280)  
print("Name- ",c.name,",Max-speed- ",c.max\_speed,",Mileage- ",c.mileage)  
c1= Bus("Volvo" , 180,230)  
print("Name- ",c1.name,",Max-speed- ",c1.max\_speed,",Mileage- ",c1.mileage)  
  
*#Program - Create a Bus child class that inherits from the Vehicle class. The default fare charge of any vehicle is seating capacity \* 100.  
# If Vehicle is Bus instance, we need to add an extra 10% on full fare as a maintenance charge.  
# So total fare for bus instance will become the final amount = total fare + 10% of the total fare.*class vehicle():  
 def \_\_init\_\_(self,name,maxspeed,mileage,capacity):  
 self.name=name  
 self.maxspeed=maxspeed  
 self.mileage=mileage  
 self.capacity=capacity  
 def fare(self):  
 print("Name- ",self.name,"Maxspeed- ",self.maxspeed,"Mileage- ",self.mileage,"Seating capacity- ",self.capacity)  
 return self.capacity\*100  
class bus(vehicle):  
 pass  
schoolbus= bus("Omni",120,260,50)  
print("Amount= ",schoolbus.fare())  
  
*#program- Write a program to determine which class a given Bus object belongs to.*class vehicle():  
 def \_\_init\_\_(self,name,maxspeed,mileage,capacity):  
 self.name=name  
 self.maxspeed=maxspeed  
 self.mileage=mileage  
 self.capacity=capacity  
class bus(vehicle):  
 pass  
  
schoolbus = bus("Maruthi",150,250,30)  
print("Name- ",schoolbus.name,",Maxspeed- ",schoolbus.maxspeed,",Mileage- ",schoolbus.mileage,",Seating capacity- ",schoolbus.capacity)  
print(type(schoolbus))

Output:

Enter no.of products you need to store: 2

Enter product code: 100

Enter product name: cooking oil

Enter cost price: 200

Enter product code: 101

Enter product name: cinthol soap

Enter cost price: 45

Stock in stores

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Product code Product name Cost price

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

100 cooking oil 200

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

101 cinthol soap 45

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enter the Quantity of product code

1002

Enter the Quantity of product code

1015

Invoice Receipt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product code | Product name | Cost price | Quantity | Total |
| 100 | Cooking oil | 200 | 2 | 400 |
| 101 | Cinthol soap | 45 | 5 | 225 |

Total Amount= 625

Enter the amount given by consumer 1000

Balance amount= 375

The Seating Capacity of a Omni is 50

Name- SWift ,Max-speed- 170 ,Mileage- 280

Name- Volvo ,Max-speed- 180 ,Mileage- 230

Name- Omni Maxspeed- 120 Mileage- 260 Seating capacity- 50

Amount= 5000

Name- Maruthi ,Maxspeed- 150 ,Mileage- 250 ,Seating capacity- 30

<class '\_\_main\_\_.bus'>