class MCart:

def \_\_init\_\_(self):

self.items = []

def add\_item(self,item\_name,qty):

item = (item\_name,qty)

self.items.append(item)

def remove\_item(self,item\_name):

for item in self.items:

if item[0] == item\_name:

self.items.remove(item)

break

def calculate\_total(self):

total = 0

for item in self.items:

total += item[1]

return total

#driver code

cart=MCart()

cart.add\_item("honey",100)

cart.add\_item("c-nuts",100)

cart.add\_item("toys",500)

print("Current Items in cart:")

for item in cart.items:

print(item[0],"-",item[1])

total\_qty = cart.calculate\_total()

print("Total Quantity:",total\_qty)

cart.remove\_item("honey")

print("\nUpdated Items in Cart after removing honey:")

for item in cart.items:

print(item[0],"-",item[1])

total\_qty = cart.calculate\_total()

print("Total Quantity:",total\_qty) Rental System

#Bike

class BikeShop:

def \_\_init\_\_(self,stock):

self.stock = stock

def ShowBike(self):

print("Total Bike :",self.stock)

def rentBike(self,quantify):

if quantify<=0:

print("please enter the positive value or greater than zero")

elif quantity>self.stock:

print("Enter the value (less than stock)")

else:

self.stock = self.stock-quantity

print("Total price :",quantity \* 100)

print("Total Bikes",self.stock)

while True:

obj=BikeShop(100)

user\_input = int(input('''1->Display Stock

2-> Rent a Bike

3->exit'''))

if user\_input ==1:

obj.ShowBike()

elif user\_input ==2:

n = int(input("Enter The rent Bike :->"))

obj.rentBike(n)

else:

break