**BIKE RENTAL**

**Code:**

*Back End Library Of Bike Rental*

import datetime

class BikeRental:

def \_\_init\_\_(self,stock=0):

"""

Our constructor class that instantiates bike rental shop.

"""

self.stock = stock

def displaystock(self):

"""

Displays the bikes currently available for rent in the shop.

"""

print("We have currently {} bikes available to rent.".format(self.stock))

return self.stock

def rentBikeOnHourlyBasis(self, n):

"""

Rents a bike on hourly basis to a customer.

"""

if n <= 0:

print("Number of bikes should be positive!")

return None

elif n > self.stock:

print("Sorry! We have currently {} bikes available to rent.".format(self.stock))

return None

else:

now = datetime.datetime.now()

print("You have rented a {} bike(s) on hourly basis today at {} hours.".format(n,now.hour))

print("You will be charged $5 for each hour per bike.")

print("We hope that you enjoy our service.")

self.stock -= n

return now

def rentBikeOnDailyBasis(self, n):

"""

Rents a bike on daily basis to a customer.

"""

if n <= 0:

print("Number of bikes should be positive!")

return None

elif n > self.stock:

print("Sorry! We have currently {} bikes available to rent.".format(self.stock))

return None

else:

now = datetime.datetime.now()

print("You have rented {} bike(s) on daily basis today at {} hours.".format(n, now.hour))

print("You will be charged $20 for each day per bike.")

print("We hope that you enjoy our service.")

self.stock -= n

return now

def rentBikeOnWeeklyBasis(self, n):

"""

Rents a bike on weekly basis to a customer.

"""

if n <= 0:

print("Number of bikes should be positive!")

return None

elif n > self.stock:

print("Sorry! We have currently {} bikes available to rent.".format(self.stock))

return None

else:

now = datetime.datetime.now()

print("You have rented {} bike(s) on weekly basis today at {} hours.".format(n, now.hour))

print("You will be charged $60 for each week per bike.")

print("We hope that you enjoy our service.")

self.stock -= n

return now

def returnBike(self, request):

"""

1. Accept a rented bike from a customer

2. Replensihes the inventory

3. Return a bill

"""

rentalTime, rentalBasis, numOfBikes = request

bill = 0

if rentalTime and rentalBasis and numOfBikes:

self.stock += numOfBikes

now = datetime.datetime.now()

rentalPeriod = now - rentalTime

print(rentalPeriod)

# hourly bill calculation

if rentalBasis == 1:

bill = round(rentalPeriod.seconds / 3600) \* 5 \* numOfBikes

# daily bill calculation

elif rentalBasis == 2:

bill = round(rentalPeriod.days) \* 20 \* numOfBikes

# weekly bill calculation

elif rentalBasis == 3:

bill = round(rentalPeriod.days / 7) \* 60 \* numOfBikes

if (3 <= numOfBikes <= 5):

print("You are eligible for Family rental promotion of 30% discount")

bill = bill \* 0.7

print("Thanks for returning your bike. Hope you enjoyed our service!")

print("That would be ${}".format(bill))

return bill

else:

print("Are you sure you rented a bike with us?")

return None

class Customer:

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

"""

Our constructor method which instantiates various customer objects.

"""

self.bikes = 0

self.rentalBasis = 0

self.rentalTime = 0

self.bill = 0

def requestBike(self):

"""

Takes a request from the customer for the number of bikes.

"""

bikes = input("How many bikes would you like to rent?")

try:

bikes = int(bikes)

except ValueError:

print("That's not a positive integer!")

return -1

if bikes < 1:

print("Invalid input. Number of bikes should be greater than zero!")

return -1

else:

self.bikes = bikes

return self.bikes

def returnBike(self):

"""

Allows customers to return their bikes to the rental shop.

"""

if self.rentalBasis and self.rentalTime and self.bikes:

return self.rentalTime, self.rentalBasis, self.bikes

else:

return 0,0,0

*Main.py Frontend For The User*

from bikerental import BikeRental, Customer

def main():

shop = BikeRental(100)

customer = Customer()

while True:

print("""

====== Bike Rental Shop =======

1. Display available bikes

2. Request a bike on hourly basis $5

3. Request a bike on daily basis $20

4. Request a bike on weekly basis $60

5. Return a bike

6. Exit

""")

choice = input("Enter choice: ")

try:

choice = int(choice)

except ValueError:

print("That's not an int!")

continue

if choice == 1:

shop.displaystock()

elif choice == 2:

customer.rentalTime = shop.rentBikeOnHourlyBasis(customer.requestBike())

customer.rentalBasis = 1

elif choice == 3:

customer.rentalTime = shop.rentBikeOnDailyBasis(customer.requestBike())

customer.rentalBasis = 2

elif choice == 4:

customer.rentalTime = shop.rentBikeOnWeeklyBasis(customer.requestBike())

customer.rentalBasis = 3

elif choice == 5:

customer.bill = shop.returnBike(customer.returnBike())

customer.rentalBasis, customer.rentalTime, customer.bikes = 0,0,0

elif choice == 6:

break

else:

print("Invalid input. Please enter number between 1-6 ")

print("Thank you for using the bike rental system.")

if \_\_name\_\_=="\_\_main\_\_":

main()

**Output:**



