**Experiment No. 15**

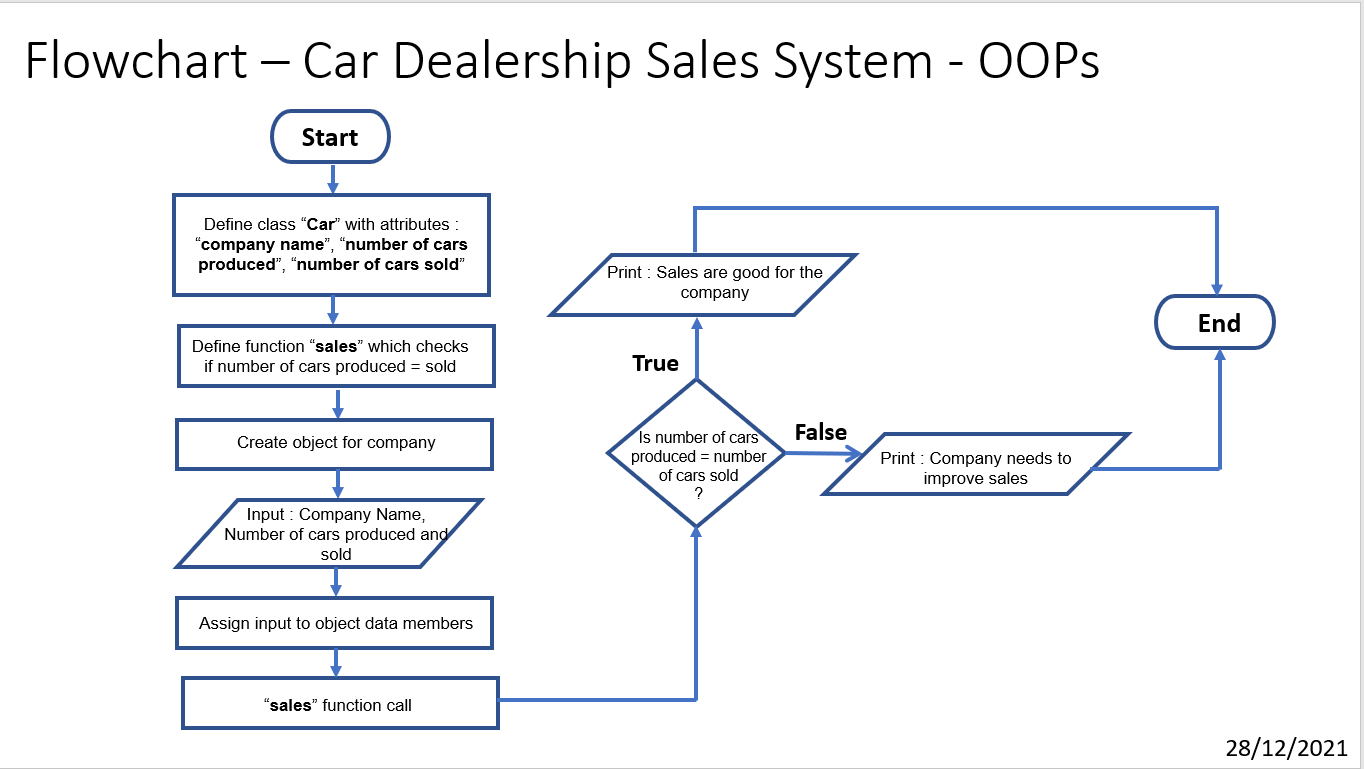
**Title:**

Write a Python Program to create a sales system for car dealerships using Object-Oriented techniques.

**Tool/Platform:**

Microsoft Word / PowerPoint and Python IDLE

**Flowchart:**



**Algorithm:**

1. Initialise program
2. Define a class “**Car**” with attributes/data members as “**company name**”, “**number of cars produced**”, “**number of cars sold**”
3. Define a function “**Sales**” in which it is checked if “number of cars produced = number of cars sold”
4. If true, “ the sales are good”, else, “sales need to be improved”
5. Create objects for each company with attributes as **company name**”, “**number of cars produced**”, “**number of cars sold**”
6. Perform function call (the Sales function defined in the class) for the object created
7. End the program

**Source Code:**

#Class - Sales System for Cars

#Date 28/12/2021

class Car():

company = " "

produced\_no = 0

sold\_no = 0

def sales(self):

if self.sold\_no == self.produced\_no:

print(f'Sales of {self.sold\_no} out of {self.produced\_no} manufactured cars by {self.company}. The company is performing well.')

else:

print(f'Sales of {self.sold\_no} out of {self.produced\_no} manufactured cars are not enough. Sales should be improved by {self.company}')

tata = Car()

tata.company = "TATA"

tata.produced\_no = 500

tata.sold\_no = 400

tata.sales()

print()

mahindra = Car()

mahindra.company = "MAHINDRA"

mahindra.produced\_no = 500

mahindra.sold\_no = 500

mahindra.sales()

print()

honda = Car()

honda.company = "HONDA"

honda.produced\_no = 500

honda.sold\_no = 500

honda.sales()

print()

kia = Car()

kia.company = "KIA"

kia.produced\_no = 400

kia.sold\_no = 400

kia.sales()

print()

toyota = Car()

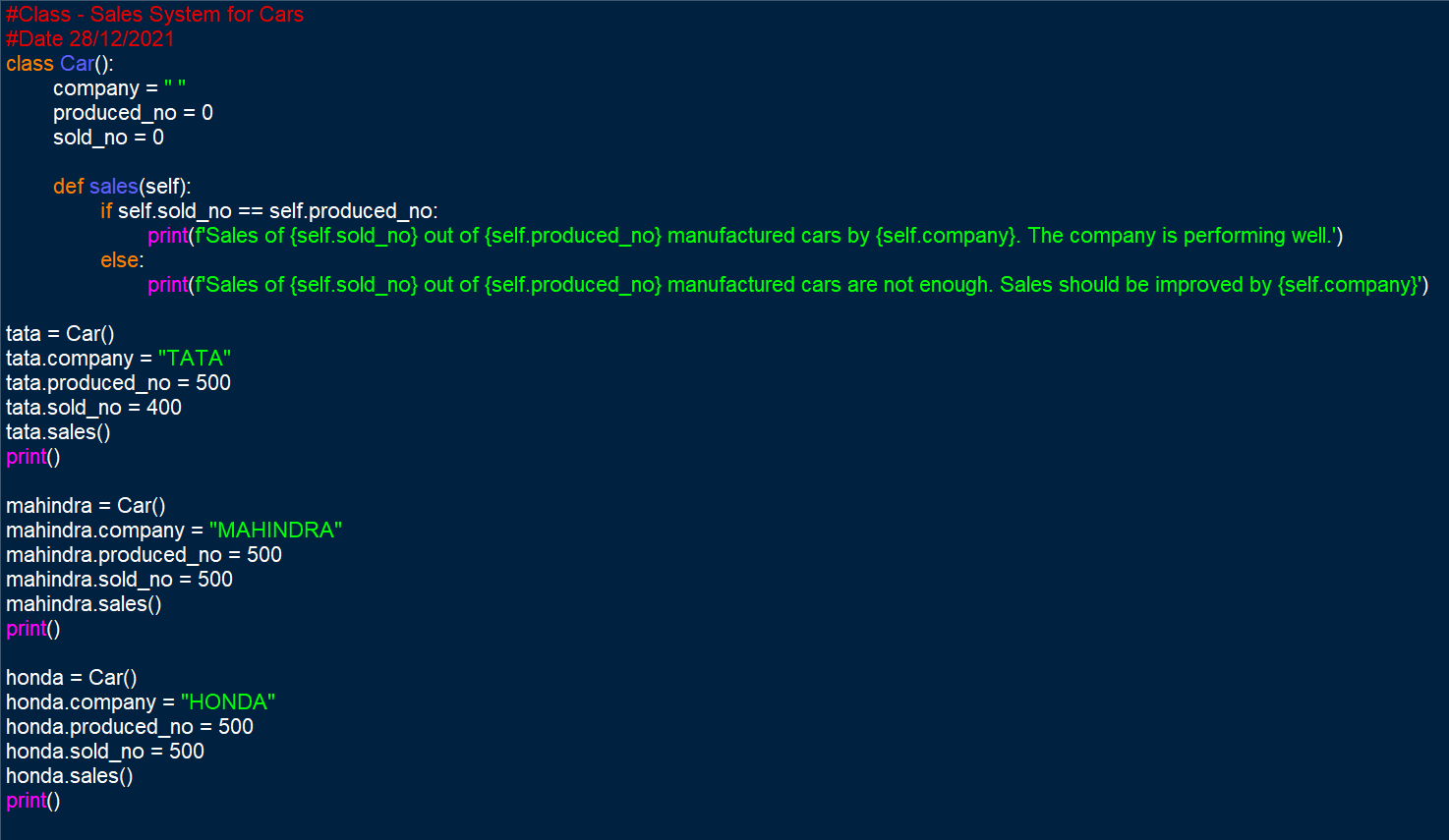
toyota.company = "TOYOTA"

toyota.produced\_no = 400

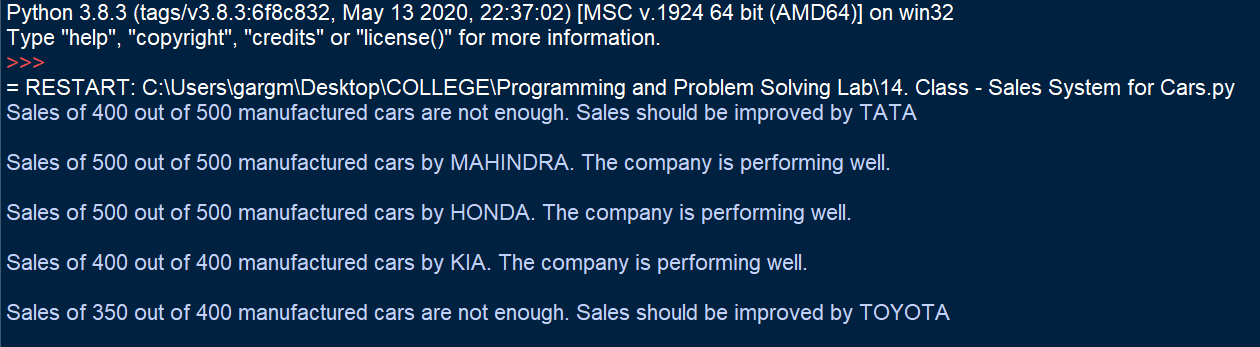
toyota.sold\_no = 350

toyota.sales()

print()

**Output Screenshot:**



**Learning Outcome:**

The following program helped in understanding the object-oriented techniques used in python programming. It helped in understanding the syntax used in OOPs and how to implement it. It helped in learning the class, object and function definitions and calls.

The following program was useful to understand the need, applications and benefits of using OOPs in python programming.