Week3 Set3

Q1

Design a program of class Car with some attributes and its object to print its attributes.

In [2]:
```
class Car:
    def __init__(self, company, color, size):
        self.company = company
        self.color = color
        self.size = size
    def display(self):
        print('The name of the car\'s company is ' + self.company)
        print('The color of the car is ' + self.color)
        print('The size of the car is ' + self.size)

car1 = Car('Honda', 'royal blue', 'large')
car2 = Car('Mercedes', 'white', 'small')

car1.display()
print()
car2.display()
```

```
The name of the car's company is Honda
The color of the car is royal blue
The size of the car is large

The name of the car's company is Mercedes
The color of the car is white
The size of the car is small
```

Q2

Develop a program of class Room with attributes length, breadth and height and its object room1 and room2 to calculate the area and volume of a room using function.

In [3]:
```
class Room:
    def __init__(self, length, breadth, height):
        self.length = length
        self.breadth = breadth
        self.height = height
    def area(self):
        return 2*(self.length*self.breadth + self.breadth*self.height +
self.height*self.length)
    def volume(self):
        return self.length*self.breadth*self.height

room1 = Room(200, 250, 210)
room2 = Room(100, 625, 190)

print('The volume of room1 is: ' + str(room1.volume()))
print('The area of room1 is: ' + str(room1.area()))

print()
print('The volume of room1 is: ' + str(room2.volume()))
print('The area of room1 is: ' + str(room2.area()))
```

```
The volume of room1 is: 10500000
The area of room1 is: 289000

The volume of room1 is: 11875000
The area of room1 is: 400500
```

Q3

A farmer wants to build a wooden fence around a rectangular field. He measures the length and the width of the field and decides how high the fence should be. He alsos decides how wide the space between each board of the fence should be.Each board is 10cm wide. Help him with a object oriented program that calculates the total length of all boards required to be bought.

In [5]:
```
class Fence:
    def __init__(self, length, breadth, space, height):
        self.length = length
        self.breadth = breadth
        self.space = space
        self.height = height
    def lengthOfBoard(self):
        perimeter = 4*(self.length+self.breadth)
        noFences = perimeter / (10+self.space)
        return noFences * self.height

field1 = Fence(1000, 2000, 5, 100)
print('The total length of fences required to be bought is: ' + str(int(
field1.lengthOfBoard())))
```

```
The total length of fences required to be bought is: 80000
```

Q4

Make a class Employee with a name and salary. Make a class Manager inherit from Employee. Add an instance variable, named department, of type string. Supply a method to String that prints the manager's name, department and salary. Make a class Executive inherits from Manager. Supply a method to String that prints the string "Executive" followed by the information stored in the Manager superclass object. Supply a test program that tests these classes and methods.

In [36]:
```
class Employee:
    def __init__(self, name, salary):
        self.name = name
        self.salary = salary

class Manager(Employee):
    def __init__(self, name, salary, department):
        super().__init__(name, salary)
        self.department = department
    def display(self):
        print('Name of the manager: ' + self.name)
        print('Department of the manager: ' + self.department)
        print('Salary of the manager: ' + str(self.salary))

class Executive(Manager):
    def __init__(self, name, salary, department):
        super().__init__(name, salary, department)
    def printExecutive(self):
        print('Executive')
        self.display()



employee1 = Employee('Sanju', 150000)
manager1 = Manager('Sanju', 150000, 'IT')

manager1.display()

executive1 = Executive('Mani', 50000, 'Web Dev')
print()
executive1.printExecutive()
```

```
Name of the manager: Sanju
Department of the manager: IT
Salary of the manager: 150000

Executive
Name of the manager: Mani
Department of the manager: Web Dev
Salary of the manager: 50000
```

Q5

Imagine a tollbooth with a class called toll Booth. The two data items are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both these to 0. A member function called payingCar ( ) increments the car total and adds 0.50 to the cash total. Another function, called nopayCar ( ), increments the car total but adds nothing to the cash total. Finally, a member function called displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car.

In [37]:
```
class TollBooth:
    def __init__(self):
        self.carNos = 0
        self.money = 0.0
    def payingCar(self):
        self.carNos += 1
        self.money += 0.50
    def nopayCar(self):
        self.carNos += 1
    def display(self):
        print('The total number of cars is: ' + str(self.carNos))
        print('The total amount of money collected is: ' + str(self.mone
y))

tollBooth1 = TollBooth()
tollBooth1.payingCar()
tollBooth1.payingCar()
tollBooth1.nopayCar()
tollBooth1.payingCar()
tollBooth1.nopayCar()
tollBooth1.display()
```

```
The total number of cars is: 5
The total amount of money collected is: 1.5
```