In [ ]:

```python
# Python program to demonstrate
# single inheritance

# Base class
class car:
    def func1(self):
        print("This function is in Parent class.")

# Derived class


class Toyota(car):
    def func2(self):
        print("This function is in child class.")


# Driver's code
object = Toyota()
object.func1()
object.func2()
```

```
This function is in Parent class.
This function is in child class.
```

In [ ]:

```python
# Python program to demonstrate
# multiple inheritance

# Base class1
class car:
    carname = ""

    def car(self):
        print(self.carname)

# Base class2


class Manufacturer:
    manufacturername = ""

    def manufacturer(self):
        print(self.manufacturername)

# Derived class


class company(car, Manufacturer):
    def car(self):
        print("Manufacturer :", self.manufacturername)
        print("Car Name :", self.carname)


# Driver's code
s1 = company()
s1.manufacturername = "DEEPAK"
s1.carname = "BMW"
s1.car()
```

```
Manufacturer : DEEPAK
Car Name : BMW
```

In [ ]:

```
# Python program to demonstrate
# multilevel inheritance

# Base class


class company:

    def __init__(self, companyname):
        self.companyname = companyname

# Intermediate class


class car(company):
    def __init__(self, carname, companyname):
        self.carname = carname

        # invoking constructor of Grandfather class
        company.__init__(self, companyname)

# Derived class


class Types(car):
    def __init__(self, Typesname, carname, companyname):
        self.Typesname = Typesname

        # invoking constructor of Father class
        car.__init__(self, carname, companyname)

    def print_name(self):
        print('company name :', self.companyname)
        print("Car name :", self.carname)
        print("Types name :", self.Typesname)


#  Driver code
s1 = Types('Luxury', 'Innova', 'Toyota')
print(s1.companyname)
s1.print_name()
```

```
Toyota
company name : Toyota
Car name : Innova
Types name : Luxury
```

In [ ]:

```
# Python program to demonstrate
# Hierarchical inheritance


# Base class
class Manufacturer:
    def func1(self):
        print("This function is in manufacturer class.")

# Derived class1

```

```


class car1(Manufacturer):
    def func2(self):
        print("This function is in car1.")

# Derivied class2


class car2(Manufacturer):
    def func3(self):
        print("This function is in car 2.")


# Driver's code
object1 = car1()
object2 = car2()
object1.func1()
object1.func2()
object2.func1()
object2.func3()
```

```
This function is in manufacturer class.
This function is in car1.
This function is in manufacturer class.
This function is in car 2.
```

In [ ]:

```
#Using functions
def decode_encoded_string(stringsplit):
    # Split the input string using underscores
    parts = stringsplit.split('_')

    # Extract the name, domain name, and register number
    name = parts[0]
    domain_name = parts[1]
    register_number = parts[2]

    # Create a dictionary to store the extracted values
    decoded_dict = {
        "name": name,
        "domain_name": domain_name,
        "register_number": register_number
    }

    return decoded_dict

# Example encoded string: "JohnDoe_MyDomain_12345"
encoded_string = "Deepak_Carsales Management_2347220"

decoded_dictionary = decode_encoded_string(encoded_string)
print(decoded_dictionary)
```

```
{'name': 'Deepak', 'domain_name': 'Carsales Management', 'register_number': '2347
220'}
```