PRACTICAL – 3

=>INHERITANCE<=

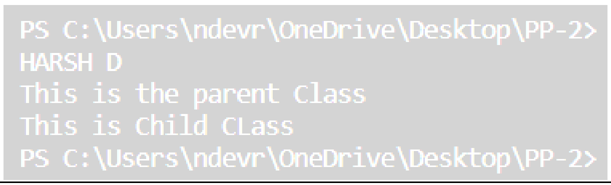
<u>PROGRAM -1</u>

AIM- WAPP FOR SINGLE INHERITANCE

CODE-

```
"""
This program demonstrates inheritance in Python
"""
print("HARSH D")
# Define the parent class
class parent:
  def fun1(self):
      """
      This method prints a message indicating that it belongs to the parent class
      """
      print("This is the parent Class")

# Define the child class that inherits from the parent class
class child(parent):
  def fun2(self):
      """
      This method prints a message indicating that it belongs to the child class
      """
      print("This is Child CLass")

# Create an instance of the child class and call its methods
object=child()
object.fun1()
object.fun2()
```

OUTPUT-



```
PS C:\Users\ndevr\OneDrive\Desktop\PP-2>
HARSH D
This is the parent Class
This is Child CLass
PS C:\Users\ndevr\OneDrive\Desktop\PP-2>
```

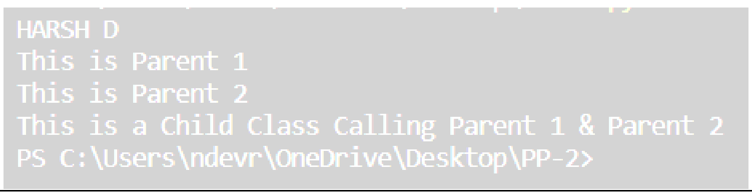
PROGRAM -2

AIM- WAPP FOR MULTIPLE INHERITANCE

CODE-

```
print("HARSH D")
class parent1():
  '''
  This is the parent1 class which has a method fun1
  '''
  def fun1(self):
    # Method to print a message
    print("This is Parent 1")

class parent2():
  '''
  This is the parent2 class which has a method fun2
  '''
  def fun2(self):
    # Method to print a message
    print("This is Parent 2")

class child(parent1,parent2):
  '''
  This is the child class inheriting from parent1 and parent2
  '''
  def fun3(self):
    # Method to print a message
    print("This is a Child Class Calling Parent 1 & Parent 2")

obj=child()
obj.fun1()
obj.fun2()
obj.fun3()
```

OUTPUT-



```
HARSH D
This is Parent 1
This is Parent 2
This is a Child Class Calling Parent 1 & Parent 2
PS C:\Users\ndevr\OneDrive\Desktop\PP-2>
```

PROGRAM -3

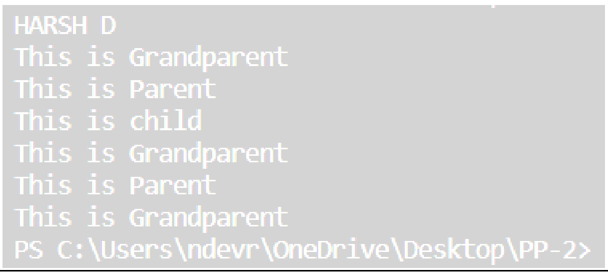
AIM- WAPP FOR MULTILEVEL INHERITANCE

CODE-

```
print("HARSH D")
class Grandparent():
    """
    This is the Grandparent class
    """
    def fun1(self):
        """
        This method prints a message for Grandparent
        """
        print("This is Grandparent")

class Parent(Grandparent):
    """
    This is the Parent class
    """
    def fun2(self):
        """
        This method prints a message for Parent
        """
        print("This is Parent")

class child(Parent):
    """
    This is the child class
    """
    def fun3(self):
        """
        This method prints a message for child
        """
        print("This is child")

# Creating objects and calling methods
obj = child()
obj.fun1()
obj.fun2()
obj.fun3()

object = Parent()
object.fun1()
object.fun2()
```

```
object1 = Grandparent()
object1.fun1()
```

OUTPUT-



```
HARSH D
This is Grandparent
This is Parent
This is child
This is Grandparent
This is Parent
This is Grandparent
PS C:\Users\ndevr\OneDrive\Desktop\PP-2>
```

PROGRAM -4

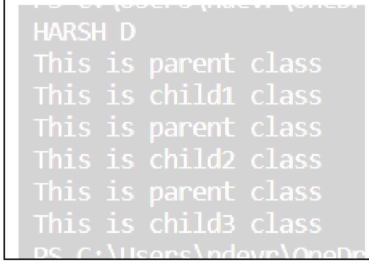
AIM- WAPP FOR **Hierarchical Inheritance:** INHERITANCE

CODE-

```
print("HARSH D")
class Parent():
  def fun1(self):
    """
    This is the fun1 method of the Parent class
    """
    print("This is parent class")

class Child1(Parent):
  def fun2(self):
    """
    This is the fun2 method of the Child1 class
    """
    print("This is child1 class")

class Child2(Parent):
  def fun3(self):
     """
     This is the fun3 method of the Child2 class
     """
     print("This is child2 class")

class Child3(Parent):
  def fun4(self):
    """
    This is the fun4 method of the Child3 class
    """
    print("This is child3 class")
```

```
object=Child1()
object.fun1()
object.fun2()

object=Child2()
object.fun1()
object.fun3()

object=Child3()
object.fun1()
object.fun4()
```

OUTPUT-



```
HARSH D
This is parent class
This is child1 class
This is parent class
This is child2 class
This is parent class
This is child3 class
PS C:\Users\ndevr\OneDr
```

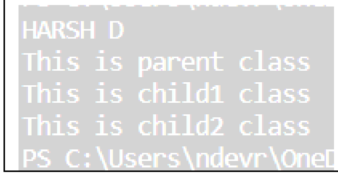
PROGRAM -5

AIM- WAPP FOR **Hybrid Inheritance:** INHERITANCE

CODE-

```
# This code defines a Parent class and multiple Child classes that inherit from the Parent
class.
print("HARSH D")

class Parent:
  def fun1(self):
    """
    This method represents the functionality of the parent class.
    """
    print("This is parent class")

class Child1(Parent):
  def fun2(self):
    """
    This method represents the functionality of the child1 class.
    """
    print("This is child1 class")

class Child2(Parent):
  def fun3(self):
    """
    This method represents the functionality of the child2 class.
    """
    print("This is child2 class")

class Child3(Parent):
  def fun4(self):
    """
    This method represents the functionality of the child3 class.
    """
    print("This is child3 class")

class HybridChild(Child1, Child2):
  """
  This class inherits from both Child1 and Child2 classes.
  """
  pass

object = HybridChild()
object.fun1()
object.fun2()
object.fun3()
```

OUTPUT-



```
HARSH D
This is parent class
This is child1 class
This is child2 class
PS C:\Users\ndevr\OneD
```