

Assignment 4 – Inheritance

Problem Statement:

You work in XYZ Corporation as a Data Analyst. Your corporation has told you to work with the inheritance of the classes.

Tasks To Be Performed:

1. Create a class named parent_Class and inside the class, initialize a global variable num as 10
 - a. Create another class named child_Class and this class should be inherited from the parent class.
 - b. Now create an object for the child_Class and with the help of child_Class object, display the value of 'num'.
2. Create three classes named A, B and C
 - a. Inside the A class, create a constructor. Inside the constructor, initialize 2 global variables name and age.
 - b. After initializing the global variables inside the constructor, now create a function named 'details' and that function should return the 'name' variable.
 - c. Inside the B class, create a constructor. Inside the constructor, initialize 2 global variables name and id.
 - d. After initializing the global variables inside the constructor, now create a function named 'details' and that function should return the 'name' variable.
 - e. The C class should inherit from class A, and B. Inside the class C, create a constructor, and inside the constructor, call the constructor of class A.
 - f. Now, create a method inside the class C, as get_details, and this function should return the value of name.
 - g. Atlast, create an object of class C, and with the help of the object, call the get_details().
3. Create a class named 'Sub1', inside the class, generate a user defined function named 'first' and inside the function, pass the following statement in the print()- 'This is the first function from Sub 1 class'.
 - a. Now create another class named 'Sub2', and inside the class, create a function named 'second', and pass the following message in the print()- 'This is the second function from the Sub 2 class'.
 - b. After that, create another class named 'Super' and inside that class, create a method named 'final', and pass the below message in the print()- 'This is the final method from the super class'.
 - c. Now, create an object for the Super class and call all the 3 user defined methods, i.e., first(), second(), and final().
4. Create a class named 'Parent', and inside the class, create a function named 'fun1' and pass the following message in the print()- 'This is the message from the fun1'.
 - a. Now create a class named 'Child1' and inside the class, create a method named 'fun2' and pass the following message in the print()- 'This is the message from the fun2'.
 - b. After that, create another class named 'Child2' and inside the class, create a method named 'fun3' and pass the following message in the print()- 'This is the message from the fun3'.
 - c. Now, create an object of Child2 class and with the help of the object, call the 'fun1' method from the 'Parent' class.
5. Create a class named 'Parent', and inside the class, create a function named 'fun1' and pass the following message in the print()- 'This is the message from the fun1'.
 - a. Now create a class named 'Child' and inside the class, create a method named 'fun2' and pass the following message in the print()- 'This is the message from the fun2'.
 - b. After that, create another class named 'Hybrid' and inside the class, create a method named 'fun3' and pass the following message in the print()- 'This is the message from the fun3'.
 - c. Now create an object of Hybrid class and with the help of the object, call the 'fun1', 'fun2' and 'fun3' methods.

```
In [4]: class parent_Class:
        global num
        num = 10

        class child_Class(parent_Class):
            def __init__(self):
                self.num = num

obj = child_Class()
obj.num
```

Out[4]: 10

```
In [12]: class A:
        def __init__(self):
            global name, age
            name, age = 'Sam', 28
        def details(self):
            self.name = name
            return(self.name)

        class B:
            def __init__(self):
                global name, id
                name, id = 'Sid', 82
            def details(self):
                self.name = name
                return(self.name)

        class C(A,B):
            def __init__(self):
                A.__init__(self)
            def get_details(self):
                self.details()
                return(self.name)

obj = C()
obj.get_details()
```

Out[12]: 'Sam'

```
In [14]: class Sub1:
        def first(self):
            print('This is the first function from Sub 1 class')

        class Sub2:
            def second(self):
                print('This is the second function from the Sub 2 class')

        class Super(Sub1, Sub2):
            def final(self):
                print('This is the final method from the super class')

obj = Super()
obj.first()
obj.second()
obj.final()
```

This is the first function from Sub 1 class
This is the second function from the Sub 2 class
This is the final method from the super class

```
In [15]: class Parent:
        def fun1(self):
            print('This is the message from the fun1')

        class Child1(Parent):
            def fun1(self):
                print('This is the message from the fun2')

        class Child2(Parent):
            def fun3(self):
                print('This is the message from the fun3')

obj = Child2()
obj.fun1()
```

This is the message from the fun1

```
In [18]: class Parent:
        def fun1(self):
            print('This is the message from the fun1')

        class Child(Parent):
            def fun2(self):
                print('This is the message from the fun2')

        class Hybrid(Child):
            def fun3(self):
                print('This is the message from the fun3')

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

This is the message from the fun1
This is the message from the fun2
This is the message from the fun3

In []: