

Practical 9

Aim: Consider an example of declaring the examination result. Design three classes: Student, Exam, and Result. The Student class has data members such as those representing rollNumber, Name, etc. Create the class Exam by inheriting Student class. The Exam class adds fields representing the marks scored in six subjects. Derive Result from the Exam class, and it has its own fields such as total_marks. Write an interactive program to model this relationship.

s/w: PyCharm Community edition 2021.3

Theory:

Inheritance is defining a new class with some modification from the existing class. The new class is called derived/child class and the one from which it inherits is called base/parents class.

Syntax:

```
class <base class>(<child class>):
```

```
    //body of base class
```

```
Class <child class>(<base class>):
```

```
    //body of child class
```

Program:

```
# Dhwani Suthar - 20CE140
# Consider an example of declaring the examination result. Design three
# classes: Student, Exam, and Result.
# The Student class has data members such as those representing rollNumber,
# Name, etc. Create the class Exam by
# inheriting Student class. The Exam class adds fields representing the
# marks scored in six subjects.
# Derive Result from the Exam class, and it has its own fields such as
# total_marks. Write an interactive program to
# model this relationship.
```

```
class Student:
    def __init__(self, rollno, name):
        self.rollno = rollno
        self.name = name

    def printing(self):
        print(f"Roll No: {self.rollno}")
        print(f"Name: {self.name}")

class Exam(Student):
    def __init__(self, rollno, name, subject):
        super().__init__(rollno, name)
```

```
        self.subject = subject

    def display(self):
        super().display()
        for i in range(len(self.subject)):
            print(f"Subject {i+1} Marks: {self.subject[i]}")

class Result(Exam):
    total_marks = 0

    def __init__(self, rollno, name, subject):
        super().__init__(rollno, name, subject)
        self.total_marks = sum(subject)

    def display(self):
        super().display()
        print(f"Total Marks: {self.total_Marks}")

if __name__ == "__main__":
    student = Student(1, "Dhwani")
    student.printing()
    print()

    exam = Exam(2, "Dharmik", [10, 20, 30])
    exam.printing()
    print()

    result = Result(3, "Sanjay", [40, 50, 60])
    result.printing()
    print()

    result = Result(4, "Parul", [70, 80, 90, 100])
    result.printing()
    print()
```

```
## Dhwani Suthar - 20CE140  
# Consider an example of declaring the examination result. Design three classes: Student, Exam, and Result.  
# The Student class has data members such as those representing rollNumber, Name, etc. Create the class Exam by  
# inheriting Student class. The Exam class adds fields representing the marks scored in six subjects.  
# Derive Result from the Exam class, and it has its own fields such as total_marks. Write an interactive program to  
# model this relationship.
```

```
class Student:  
    def __init__(self, rollno, name):  
        self.rollno = rollno  
        self.name = name  
  
    def printing(self):  
        print(f"Roll No: {self.rollno}")  
        print(f"Name: {self.name}")  
  
class Exam(Student):  
    def __init__(self, rollno, name, subject):  
        super().__init__(rollno, name)  
        self.subject = subject  
  
    def display(self):  
        super().display()  
        for i in range(len(self.subject)):  
            print(f"Subject {i+1} Marks: {self.subject[i]}")  
  
class Result(Exam):  
    total_marks = 0  
  
    def __init__(self, rollno, name, subject):  
        super().__init__(rollno, name, subject)  
        self.total_marks = sum(subject)  
  
    def display(self):  
        super().display()  
        print(f"Total Marks: {self.total_Marks}")  
  
if __name__ == "__main__":  
    student = Student(1, "Dhwani")  
    student.printing()  
    print()  
  
    exam = Exam(2, "Dharmik", [10, 20, 30])  
    exam.printing()  
    print()  
  
    result = Result(3, "Sanjay", [40, 50, 60])  
    result.printing()  
    print()  
  
    result = Result(4, "Parul", [70, 80, 90, 100])  
    result.printing()  
    print()
```

Output:

Roll No: 1
Name: Dhwani

Roll No: 2
Name: Dharmik

Roll No: 3
Name: Sanjay

Roll No: 4
Name: Parul

Process finished with exit code 0

Conclusion:

1. Parent class is the class being inherited from, also called base class.
2. Child class is the class that inherits from another class, also called derived class.

GitHub: [Upload files · Dhwani-S/Python-PIP \(github.com\)](#)