**Activity 9**

Write a program to implement an employee management system using classes, instances and inheritance.

class Employee:

    def \_\_init\_\_(self, name, employee\_id):

        self.name = name

        self.employee\_id = employee\_id

    def display\_info(self):

        print(f"Employee Name: {self.name}")

        print(f"Employee ID: {self.employee\_id}")

class Manager(Employee):

    def \_\_init\_\_(self, name, employee\_id, department):

        super().\_\_init\_\_(name, employee\_id)

        self.department = department

    def display\_info(self):

        super().display\_info()

        print(f"Department: {self.department}")

class Developer(Employee):

    def \_\_init\_\_(self, name, employee\_id, programming\_language):

        super().\_\_init\_\_(name, employee\_id)

        self.programming\_language = programming\_language

    def display\_info(self):

        super().display\_info()

        print(f"Programming Language: {self.programming\_language}")

# Get user input for manager

manager\_name = input("Enter manager's name: ")

manager\_id = int(input("Enter manager's employee ID: "))

manager\_department = input("Enter manager's department: ")

# Get user input for developer

developer\_name = input("Enter developer's name: ")

developer\_id = int(input("Enter developer's employee ID: "))

programming\_language = input("Enter developer's programming language: ")

# Create instances of Manager and Developer classes

manager = Manager(manager\_name, manager\_id, manager\_department)

developer = Developer(developer\_name, developer\_id, programming\_language)

# Display employee information

print("\nManager Information:")

manager.display\_info()

print("\nDeveloper Information:")

developer.display\_info()