Assignment: Employee Management using Inheritance

Create a program that models an organization's employee hierarchy using inheritance.

Requirements:

Base Class Employee

Attributes:

emp_id

name

salary

Methods:

display_info() prints basic employee details.

calculate_bonus() returns 5% of salary (default bonus).

---

define below derived classes;

Derived Class 1 Developer

Inherits from Employee

Additional Attribute:

programming_language

Methods:

Override calculate_bonus() 10% of salary.

Add show_projects() displays assigned projects.

Derived Class 2 Manager

Inherits from Employee

Additional Attributes:

team_size

department

Methods:

Override calculate_bonus() 15% of salary.

assign_task() prints a message about task assignment.

Derived Class 3 Intern

Inherits from Employee

Additional Attribute:

duration (in months)

Methods:

Override calculate_bonus() Fixed ₹1000 bonus.

extend_internship() increases duration.

==========

USING ABOVE ALL, IMPLEMENT INHERITANCE AND POLYMORPHISM and Generate BELOW EXPECTED OUTPOUT:

Employee ID: 101, Name: Neha, Salary: ₹80000
Bonus: ₹8000.0

Employee ID: 102, Name: Raj, Salary: ₹120000
Bonus: ₹18000.0

Employee ID: 103, Name: Amit, Salary: ₹15000
Bonus: ₹1000

Neha is working on Python-based backend projects.
Manager Raj assigned tasks to 10 team members in IT.
Internship extended. New duration: 8 months.

```
class Employee:
    def __init__(self, emp_id, name, salary):
```

```
        self.name=name
        self.emp_id=emp_id
        self.salary=salary
    def display_info(self):
        print(f"Employee ID: {self.emp_id}, Name: {self.name}, Salary: ${self.salary}")
    def calculate_bonus(self):
        return self.salary*0.05

class Developer(Employee):
    def __init__(self, emp_id, name, salary, programming_language, projects):
        super().__init__(emp_id, name, salary)
        self.programming_language = programming_language
        self.projects=projects
    def calculate_bonus(self):
        return self.salary * 0.10
    def show_projects(self):
        print(self.projects)

class Manager(Employee):
    def __init__(self, emp_id, name, salary, team_size, department):
        super().__init__(emp_id, name, salary)
        self.team_size = team_size
        self.department = department
    def calculate_bonus(self):
        return self.salary * 0.15
    def assign_task(self):
        print("Task assigned")


class Intern(Employee):
    def __init__(self, emp_id, name, salary, duration):
        super().__init__(emp_id, name, salary)
        self.duration = duration
    def calculate_bonus(self):
```

```
        return 1000
    def extend_internship(self, months):
        self.duration += months
        print(f"Internship duration extended. New duration is {self.duration} months.")

Neha = Developer(emp_id=101, name="Neha", salary=80000,
programming_language="Python", projects='backend projects')

Raj = Manager(emp_id=102, name="Raj", salary=120000, team_size=10,
department="IT")

Amit = Intern(emp_id=103, name="Amit", salary=15000, duration=6)

employees = [Neha, Raj, Amit]

for emp in employees:
    emp.display_info()
    bonus_amount = emp.calculate_bonus()
    print(f"Bonus: ₹{bonus_amount}")
    print("----------------------------------------")
```