

Assignment No. 06

➤ **TITLE :** Implementation of Inheritance for Employee Payroll System in C++.

➤ **Problem Statement :**

Design and develop an inheritance-based system where:

- The Employee class contains data members: Emp_name, Emp_id, Address, Mail_id, and Mobile_no.
- The classes Programmer, Team Lead, Assistant Project Manager, and Project Manager are inherited from Employee.
- Each class includes Basic Pay (BP) as a member.
- Calculate allowances and deductions:
 - DA = 52% of BP
 - HRA = 27% of BP
 - PF = 12% of BP
 - Staff Club Fund = 0.1% of BP
- Generate a pay slip with Gross Salary and Net Salary.

➤ **Salary Computation –**

$$DA = 0.52 \times BP$$

$$HRA = 0.27 \times BP$$

$$PF = 0.12 \times BP$$

$$Staff_Fund = 0.001 \times BP$$

$$Gross_Salary = BP + DA + HRA$$

$$Net_Salary = Gross_Salary - (PF + Staff_Fund)$$

➤ **Algorithm –**

1. Start.
2. Define base class Employee with attributes: Emp_name, Emp_id, Address, Mail_id, Mobile_no.
3. Derive classes Programmer, TeamLead, AsstProjectManager, ProjectManager from Employee.
4. Add data member BasicPay and functions for salary calculation in derived classes.
5. Input employee details and BP.
6. Compute DA, HRA, PF, Staff Club Fund, Gross Salary, and Net Salary.
7. Display a pay slip with all details.
8. End.

➤ Code –

```

#include <iostream>
#include <string>
using namespace std;

class Employee {
public:
    int emp_id;
    string emp_name, emp_addr, emp_mail;
    long long emp_phone;

    void getDetails() {
        cin.ignore();
        cout << "Enter Employee Name: ";
        getline(cin, emp_name);
        cout << "Enter Employee ID: ";
        cin >> emp_id;
        cin.ignore();
        cout << "Enter Address: ";
        getline(cin, emp_addr);
        cout << "Enter Mail ID: ";
        getline(cin, emp_mail);
        cout << "Enter Mobile Number: ";
        cin >> emp_phone;
    }

    void displayDetails() {
        cout << "\nEmployee Details:\n";
        cout << "Name: " << emp_name << endl;
        cout << "ID: " << emp_id << endl;
        cout << "Address: " << emp_addr << endl;
        cout << "Mail ID: " << emp_mail << endl;
        cout << "Mobile Number: " << emp_phone << endl;
    }
};

class Salary : public Employee {
public:
    float BasicPay, DA, HRA, PF, StaffClubFund, GrossSalary, NetSalary;

    virtual void getSalary() {
        cout << "Enter Basic Pay Salary: ";
        cin >> BasicPay;
    }

    virtual void calculate() {
        DA = 0.52 * BasicPay;
        HRA = 0.27 * BasicPay;
        PF = 0.12 * BasicPay;
        StaffClubFund = 0.001 * BasicPay;
        GrossSalary = BasicPay + DA + HRA;
        NetSalary = GrossSalary - (PF + StaffClubFund);
    }

    virtual void generatePaySlip() {
        displayDetails();
        cout << "Basic Pay: " << BasicPay << endl;
        cout << "DA: " << DA << endl;
        cout << "HRA: " << HRA << endl;
    }
};

```

```

        cout << "PF: " << PF << endl;
        cout << "Staff Club Fund: " << StaffClubFund << endl;
        cout << "Gross Salary: " << GrossSalary << endl;
        cout << "Net Salary: " << NetSalary << endl;
    }
};

class Programmer : public Salary {
public:
    void calculate() override {
        DA = 0.40 * BasicPay;
        HRA = 0.20 * BasicPay;
        PF = 0.10 * BasicPay;
        StaffClubFund = 0.001 * BasicPay;
        GrossSalary = BasicPay + DA + HRA;
        NetSalary = GrossSalary - (PF + StaffClubFund);
    }
    void generatePaySlip() override {
        cout << "\nPay Slip: Programmer\n";
        Salary::generatePaySlip();
    }
};

class TeamLead : public Salary {
public:
    void calculate() override {
        DA = 0.45 * BasicPay;
        HRA = 0.22 * BasicPay;
        PF = 0.11 * BasicPay;
        StaffClubFund = 0.001 * BasicPay;
        GrossSalary = BasicPay + DA + HRA;
        NetSalary = GrossSalary - (PF + StaffClubFund);
    }
    void generatePaySlip() override {
        cout << "\nPay Slip: Team Lead\n";
        Salary::generatePaySlip();
    }
};

class AsstProjectManager : public Salary {
public:
    void generatePaySlip() override {
        cout << "\nPay Slip: Assistant Project Manager\n";
        Salary::generatePaySlip();
    }
};

class ProjectManager : public Salary {
public:
    void calculate() override {
        DA = 0.55 * BasicPay;
        HRA = 0.30 * BasicPay;
        PF = 0.12 * BasicPay;
        StaffClubFund = 0.001 * BasicPay;
        GrossSalary = BasicPay + DA + HRA;
        NetSalary = GrossSalary - (PF + StaffClubFund);
    }
    void generatePaySlip() override {
        cout << "\nPay Slip: Project Manager\n";
        Salary::generatePaySlip();
    }
};

```

```

    }
};

int main() {
    int choice;
    cout << "Select Employee Type: \n1. Programmer\n2. Team Lead\n3. Assistant Project
Manager\n4. Project Manager\n";
    cin >> choice;

    Salary* emp = nullptr;

    switch (choice) {
        case 1: emp = new Programmer(); break;
        case 2: emp = new TeamLead(); break;
        case 3: emp = new AsstProjectManager(); break;
        case 4: emp = new ProjectManager(); break;
        default: cout << "Invalid choice!" << endl; return 0;
    }

    emp->getDetails();

    emp->getSalary();
    emp->calculate();
    emp->generatePaySlip();

    delete emp;
    return 0;
}

```

➤ Output –

Select Employee Type:
 1. Programmer
 2. Team Lead
 3. Assistant Project Manager
 4. Project Manager
 3
 Enter Employee Name: Apurv
 Enter Employee ID: 123
 Enter Address: Pune
 Enter Mail ID: apurv@gmail.com
 Enter Mobile Number: 7058229202
 Enter Basic Pay Salary: 75000
 Pay Slip: Assistant Project Manager

Employee Details:
 Name: Apurv
 ID: 123
 Address: Pune
 Mail ID: apurv@gmail.com
 Mobile Number: 7058229202
 Basic Pay: 75000
 DA: 39000
 HRA: 20250
 PF: 9000
 Staff Club Fund: 75
 Gross Salary: 134250
 Net Salary: 125175

Conclusion –

In this assignment, we implemented inheritance and calculated the salary components on an employee salary payroll management system.