

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

```
class Employee {
public:
    string empID;
    string name;
    int hoursWorked;
    float hourlyRate;
```

```
    Employee(string id="", string n="", int h=0, float r=0.0) {
        empID = id;
        name = n;
        hoursWorked = h;
        hourlyRate = r;
    }
```

```
    float calculateSalary() {
        return hoursWorked * hourlyRate;
    }
```

```

        return hoursWorked * hourlyRate;
    }

    void showSlip() {
        cout << "\n--- Salary Slip ---\n";
        cout << "Employee ID: " << empID << endl;
        cout << "Name      : " << name << endl;
        cout << "Hours Worked: " << hoursWorked << endl;
        cout << "Hourly Rate : " << hourlyRate << endl;
        cout << "Total Salary: " << calculateSalary() << endl;
        cout << "-----\n";
    }
};

```

```

class PayrollSystem {
private:
    Employee employees[100]; // fixed size array
    int count;               // total employees

public:
    PayrollSystem() {
        count = 0;
    }
};

```


public:

```
PayrollSystem() {  
    count = 0;  
}
```

```
void addEmployee(Employee e) {  
    if (count < 100) {  
        employees[count] = e;  
        count++;  
        cout << "Employee added successfully.\n";  
    } else {  
        cout << "Employee list is full!\n";  
    }  
}
```

```
Employee* searchByID(string id) {  
    for (int i = 0; i < count; i++) {  
        if (employees[i].empID == id)  
            return &employees[i];  
    }  
    return nullptr;  
}
```

```
    }  
    return nullptr;  
}
```

```
void updateEmployee(string id) {  
    Employee* e = searchByID(id);  
    if (e) {  
        cout << "Enter new name: ";  
        getline(cin, e->name);  
        cout << "Enter new hours worked: ";  
        cin >> e->hoursWorked;  
        cout << "Enter new hourly rate: ";  
        cin >> e->hourlyRate;  
        cin.ignore();  
        cout << "Employee updated successfully.\n";  
    } else {  
        cout << "Employee not found.\n";  
    }  
}
```

```
void deleteEmployee(string id) {  
    for (int i = 0; i < count; i++) {
```



```
        employees[j] = employees[j+1];
    }
    count--;
    cout << "Employee deleted successfully.\n";
    return;
}
}
cout << "Employee not found.\n";
}

void listEmployees() {
    if (count == 0) {
        cout << "No employees found.\n";
        return;
    }
    cout << "\n--- Employee List ---\n";
    for (int i = 0; i < count; i++) {
        cout << "ID: " << employees[i].empID
            << ", Name: " << employees[i].name
            << ", Salary: " << employees[i].calculateSalary()
            << endl;
    }
}
```

```
}
```

```
void summaryReport() {  
    float totalPayroll = 0;  
    for (int i = 0; i < count; i++) {  
        totalPayroll += employees[i].calculateSalary();  
    }  
    cout << "\nTotal Payroll Amount: " << totalPayroll << endl;  
}  
};
```

```
int main() {  
    PayrollSystem ps;  
    int choice;  
  
    do {  
        cout << "\n--- Employee Payroll System ---\n";  
        cout << "1. Add Employee\n";  
        cout << "2. Update Employee\n";  
        cout << "3. Delete Employee\n";  
        cout << "4. Search Employee by ID\n";
```



```
cout << "5. List Employees with Salary\n";
cout << "6. Summary Report\n";
cout << "7. Generate Salary Slip\n";
cout << "8. Exit\n";
cout << "Enter your choice: ";
cin >> choice;
cin.ignore();
```

I

```
if (choice == 1) {
    string id, name;
    int hours;
    float rate;
    cout << "Enter Employee ID: ";
    getline(cin, id);
    cout << "Enter Name: ";
    getline(cin, name);
    cout << "Enter Hours Worked: ";
    cin >> hours;
    cout << "Enter Hourly Rate: ";
    cin >> rate;
    cin.ignore();
    Employee e(id, name, hours, rate);
```



```
Employee e(id, name, hours, rate);
ps.addEmployee(e);
}
else if (choice == 2) {
    string id;
    cout << "Enter Employee ID to update: ";
    getline(cin, id);
    ps.updateEmployee(id);
}
else if (choice == 3) {
    string id;
    cout << "Enter Employee ID to delete: ";
    getline(cin, id);
    ps.deleteEmployee(id);
}
else if (choice == 4) {
    string id;
    cout << "Enter Employee ID to search: ";
    getline(cin, id);
    Employee* e = ps.searchByID(id);
    if (e) {
        cout << "Found: " << e->name << " with salary " << e -
```



```
        ->calculateSalary() << endl;
    } else {
        cout << "Employee not found.\n";
    }
}
else if (choice == 5) {
    ps.listEmployees();
}
else if (choice == 6) {
    ps.summaryReport();
}
else if (choice == 7) {
    string id;
    cout << "Enter Employee ID for salary slip: ";
    getline(cin, id);
    Employee* e = ps.searchByID(id);
    if (e) e->showSlip();
    else cout << "Employee not found.\n";
}
} while (choice != 8);

cout << "Exiting program. Goodbye!\n";
```

```

    }
}
else if (choice == 5) {
    ps.listEmployees();
}
else if (choice == 6) {
    ps.summaryReport();
}
else if (choice == 7) {
    string id;
    cout << "Enter Employee ID for salary slip: ";
    getline(cin, id);
    Employee* e = ps.searchByID(id);
    if (e) e->showSlip();
    else cout << "Employee not found.\n";
}
} while (choice != 8);

cout << "Exiting program. Goodbye!\n";
return 0;
}

```