Reg. No: 21BCE1297 **Name: Vidhi Shah** **Date: 19/05/22**

PPS15

Aim:

Q1: Write a C++ program to create an employee database using class. Each employee should have name, emp_id and salary. Use getdata() and putdata() member functions to read and print the details of employees.

Q2: For the above database perform Linear searching and print a specific employee detail.

Procedure:

Input:

Number of employees, n
Employee details
Employee ID of employee whose specific employee detail is required

Output:

Employee details of the employee whose ID is entered

Algorithm:

Class Employee:

Step 1: Create a class Employee
Step 2: Add private data members: Name, ID and Salary
Step 3: Add public member function

1. **getdata():** Input user data for employee details
2. **putdata():** Output user data for employee details

Step 4: Add **Friend** member function

1. **linearSearch(int x, Employee array[]):** Find the details of employee having user input employee id

Main Function:

Step 1: Read number of employees n
Step 2: Create an Employee class array with n elements, database[n]
Step 3: Using for loop input data for all employees using 'getdata' function
Step 4: Input ID of employee whose details are required, x
Step 5: Call 'linearSearch' friend function with inputs as 'x' and 'database'
Step 6: Return 0

LinearSearch Friend Function:

Step 1: Use for loop to iterate through each object in the array

Step 2: Compare employee ID of each object to the ID entered

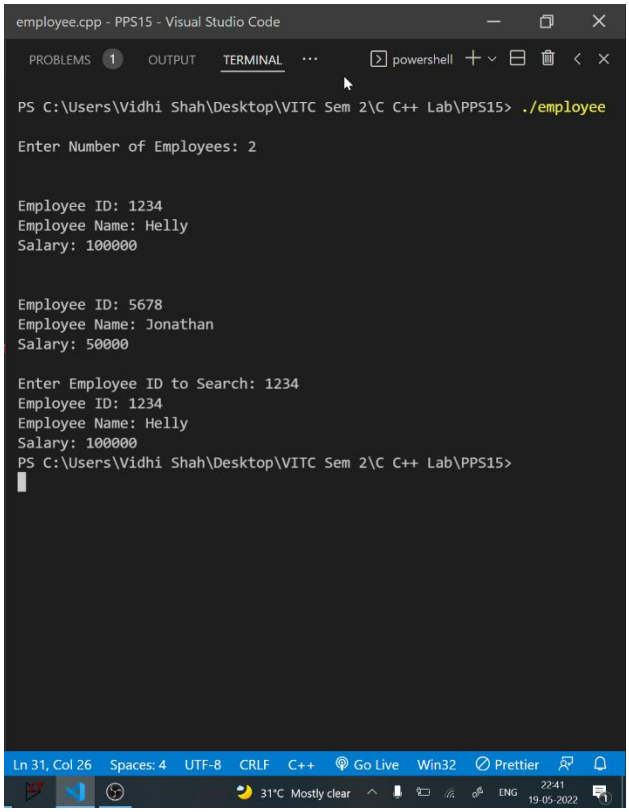
Step 3: If IDs match then print the employee details using ‘putdata’ function

Code:

```
#include <iostream>
using namespace std;
int n;
class Employee {
    char name[30];
    int emp_id, salary;

    public:
    void getdata () {
        cout<<"\n\nEmployee ID: ";
        cin>>emp_id;
        cout<<"Employee Name: ";
        cin>>name;
        cout<<"Salary: ";
        cin>>salary;
    }
    void putdata () {
        cout<<"Employee ID: "<<emp_id<<"\n";
        cout<<"Employee Name: "<<name<<"\n";
        cout<<"Salary: "<<salary<<"\n";
    }
    friend void linearSearch (int x, Employee array[]);
};

int main() {
    int i, x;
    cout<<"Enter Number of Employees: ";
    cin>>n;
    Employee database[n];

    for (i = 0; i < n; i++) {
        database[i].getdata();
    }

    cout<<"\nEnter Employee ID to Search: ";
    cin>>x;
    linearSearch(x,database);
    return 0;
}
```

```
void linearSearch (int x, Employee array[]) {
    int i, id;
    for (i = 0; i < n; i++) {
        id = array[i].emp_id;
        if (x == id) {
            array[i].putdata();
        }
    }
}
```

Output:



Q3

Aim:

Program to create a class ACCOUNTS with function read() to input sales and purchase details. Create a Friend function to print total tax to pay. Assume 4% of profit is tax.

Procedure:

Input:

Number of items, n
Purchase price for ‘n’ items
Sales price for ‘n’ items

Output:

Total tax

Algorithm:

Class Accounts:

Step 1: Create a class Accounts
Step 2: Add private data members: purchase, sales
Step 3: Add public member function

1. **read():**
    a. Enter number of items, n
    b. Using for loop enter purchase and sales price for n items

Step 4: Add **Friend** member function

1. **tax(Accounts object):** Calculate profit and tax

Main Function:

Step 1: Create an Accounts object, ‘company’
Step 2: Call read function
Step 3: Call ‘tax’ friend function with inputs as ‘company’
Step 4: Return 0

Tax Friend Function:

Step 1: profit = sales - purchase
Step 2: If profit > 0, then tax = 0.04 * profit, print tax
Step 3: Else print “Company in Loss”

Code:

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

class Accounts {
    int purchase, sales;

    public:
    void read() {
        int i, n;
        cout<<"\nEnter Number of Items: ";
        cin>>n;

        int p[n], s[n];
        purchase = 0;
        cout<<"\nEnter Purchase Price of each Item: \n";
        for (i = 0; i < n; i++) {
            cout<<"Item "<<i+1<<": ";
            cin>>p[i];
            purchase += p[i];
        }
        sales = 0;
        cout<<"\nEnter Sales Price of each Item: \n";
        for (i = 0; i < n; i++) {
            cout<<"Item "<<i+1<<": ";
            cin>>s[i];
            sales += s[i];
        }
    }

    friend void tax (Accounts company);
};

int main() {

    Accounts company;
    company.read();
    tax(company);
    return 0;
}

void tax (Accounts company) {
    int profit;
    float tax;

    profit = company.sales - company.purchase;
```

```
    if (profit > 0) {
        tax = 0.04 * profit;
        cout<<"\nTotal Tax to Pay: "<<tax;
    }
    else {
        cout<<"Company in Loss";
    }
}
```

Output: