# Assignment-2

Name - Sahil Badve
PRN - B24CE1114
Div - S.Y.B-Tech - 2
Batch - C

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

class Employee {
    char* name;
    int empID;
    char* department;
    char* dateOfJoining;
    char* contactAddress;
    long phoneNumber;
    static int count;

public:
    //Default constructor
    Employee() {
        name = new char[1];
        name[0] = '\0';
        empID = 0;
        department = new char[1];
        department[0] = '\0';
        dateOfJoining = new char[1];
        dateOfJoining[0] = '\0';
        contactAddress = new char[1];
        contactAddress[0] = '\0';
        phoneNumber = 0;
        count++;
        cout << "This is the Default Constructor" << endl;
    }

    //Parameterized constructor
    Employee(const char* n, int id, const char* dept, const char* doj, const char* address, long
phone) {
        name = new char[strlen(n)]; strcpy(name, n);
        empID = id;
        department = new char[strlen(dept)]; strcpy(department, dept);
```

```
        dateOfJoining = new char[strlen(doj)]; strcpy(dateOfJoining, doj);
        contactAddress = new char[strlen(address)]; strcpy(contactAddress, address);
        phoneNumber = phone;
        count++;
        cout << "This is the Parameterized Constructor" << endl;
    }

    //Copy constructor
    Employee(const Employee& emp) {
        name = new char[strlen(emp.name)]; strcpy(name, emp.name);
        empID = emp.empID;
        department = new char[strlen(emp.department)]; strcpy(department, emp.department);
        dateOfJoining = new char[strlen(emp.dateOfJoining)]; strcpy(dateOfJoining,
emp.dateOfJoining);
        contactAddress = new char[strlen(emp.contactAddress)]; strcpy(contactAddress,
emp.contactAddress);
        phoneNumber = emp.phoneNumber;
        count++;
    }

    inline void accept() {
        char buffer[100];

        cout << "Enter Name: ";
        cin.ignore();
        cin.getline(buffer, 100);
        delete[] name;
        name = new char[strlen(buffer)]; strcpy(name, buffer);

        cout << "Enter Employee ID: ";
        cin >> empID;

        cout << "Enter Department: ";
        cin.ignore();
        cin.getline(buffer, 100);
        delete[] department;
        department = new char[strlen(buffer)]; strcpy(department, buffer);

        cout << "Enter Date of Joining (DD/MM/YYYY): ";
        cin.getline(buffer, 100);
        delete[] dateOfJoining;
        dateOfJoining = new char[strlen(buffer)];
        strcpy(dateOfJoining, buffer);
```

```
        cout << "Enter Contact Address: ";
        cin.getline(buffer, 100);
        delete[] contactAddress;
        contactAddress = new char[strlen(buffer)];
        strcpy(contactAddress, buffer);

        cout << "Enter Telephone Number: ";
        cin >> phoneNumber;
    }

    inline void display() const {
        cout << "\nEmployee Details:\n";
        cout << "Name: " << name << endl;
        cout << "Employee ID: " << empID << endl;
        cout << "Department: " << department << endl;
        cout << "Date of Joining: " << dateOfJoining << endl;
        cout << "Contact Address: " << contactAddress << endl;
        cout << "Phone Number: " << phoneNumber << endl;
    }



    ~Employee() {
        delete[] name;
        delete[] department;
        delete[] dateOfJoining;
        delete[] contactAddress;
    }
};

int Employee::count = 0;

int main() {
    int n;
    cout << "Enter number of employees: ";
    cin >> n;

    //Default constructor
    Employee* empList = new Employee[n];

    for (int i = 0; i < n; i++) {
        cout << "\nEnter details for employee " << i + 1 << ":\n";
        empList[i].accept();
    }
```

```
    //Display all employees
    for (int i = 0; i < n; i++) {
        cout << "\n--- Employee " << i + 1 << " ---";
        empList[i].display();
    }

    //copy each employee
    for (int i = 0; i < n; i++) {
        cout << "\nCreating a copy of Employee " << i + 1 << "...\n";
        Employee empCopy(empList[i]); // Copy constructor called here
        empCopy.display();
        cout << "This is the Copy Constructor\n";
    }

    delete[] empList;
    return 0;
}
```

## OUTPUT:-

**Enter number of employees: 2**
**This is the Default Constructor**

**Enter details for employee 1:**
**Enter Name: abc**
**Enter Employee ID: 123**
**Enter Department: c**
**Enter Date of Joining (DD/MM/YYYY): 12/12/2012**
**Enter Contact Address: qwerty**
**Enter Telephone Number: 1234567890**

**Enter details for employee 2:**
**Enter Name: def**
**Enter Employee ID: 456**
**Enter Department: f**
**Enter Date of Joining (DD/MM/YYYY): 13/12/2013**
**Enter Contact Address: yuiop**
**Enter Telephone Number: 0987654321**

**--- Employee 1 ---**
**Employee Details:**
**Name: abc**
**Employee ID: 123**
**Department: c**
**Date of Joining: 12/12/2012**
**Contact Address: qwerty**
**Phone Number: 1234567890**

**--- Employee 2 ---**
**Employee Details:**
**Name: def**
**Employee ID: 456**
**Department: f**
**Date of Joining: 13/12/2013**
**Contact Address: yuiop**
**Phone Number: 987654321**

**Creating a copy of Employee 1...**

**Employee Details:**
**Name: abc**
**Employee ID: 123**
**Department: c**
**Date of Joining: 12/12/2012**
**Contact Address: qwerty**
**Phone Number: 1234567890**
**This is the Copy Constructor**

**Creating a copy of Employee 2...**

**Employee Details:**
**Name: def**
**Employee ID: 456**
**Department: f**
**Date of Joining: 13/12/2013**
**Contact Address: yuiop**
**Phone Number: 987654321**
**This is the Copy Constructor**