Name - Srushti Bhivaji Salgar
PRN - B24CE1079
SUB - Object Oriented Programming
ASSIGNMENT 2

/*Develop a program in C++ to create a database of an employee's information system containing the following fields: Name, employee ID, Department, Date of Joining, Contact address, Telephone number etc. Construct the database with suitable member functions to accept and print employee details. Make use of constructor types, destructor, static members, inline function and dynamic memory allocation using operators-new and delete.*/

CODE

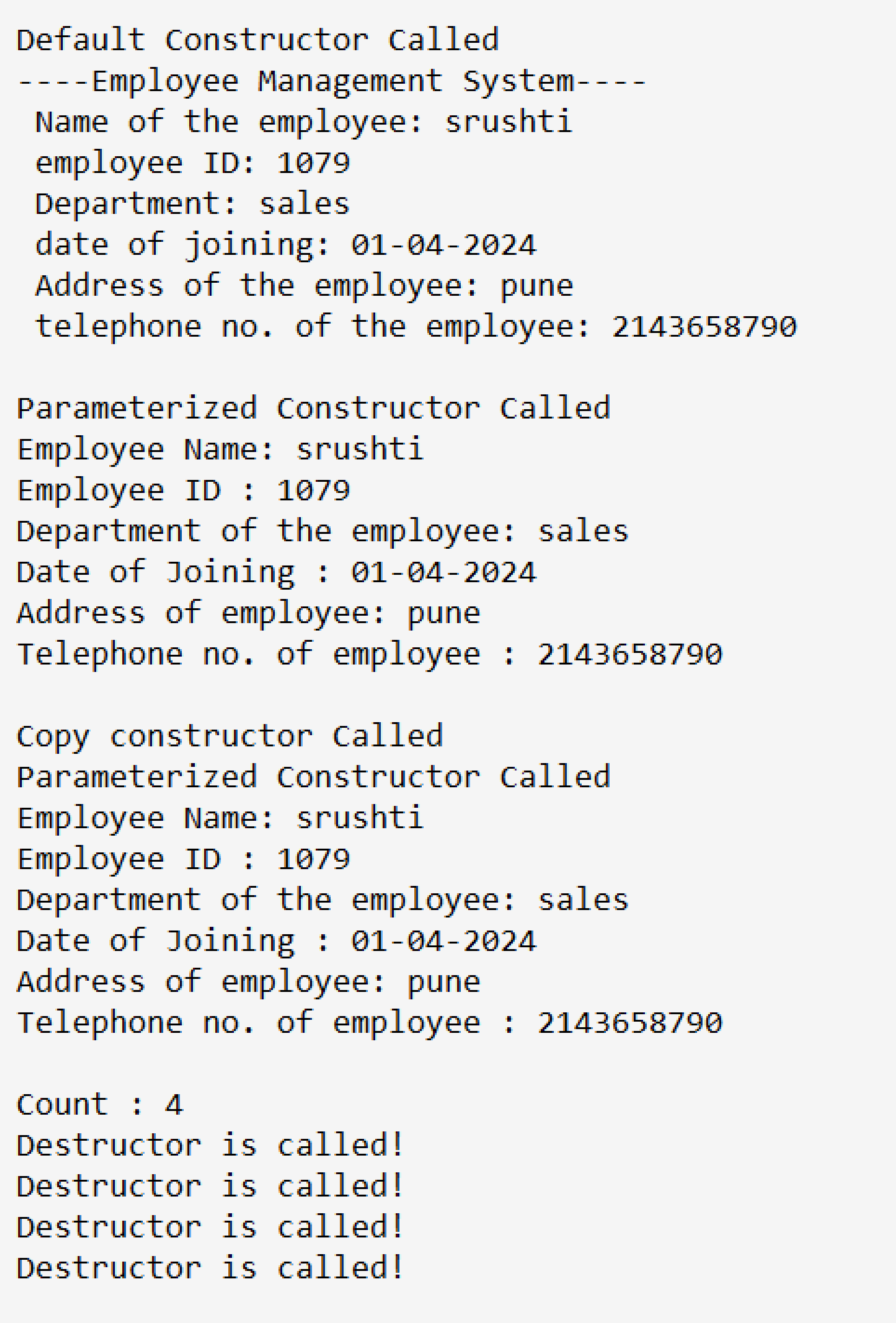
```
#include<iostream>
using namespace std;
class employee
{
private:
string name;
long int emp_id;
string dept;
string date_of_Join;
string address;
long int telephone;

static int count;
public:
inline void title(); //Inline function
employee(); //Default Constructor
employee(string n, long int id, string dep, string doj, string add, int tele);
//Parameterized Construstor
employee(employee &obj);//Copy Constructor
void display();
void display2();
~employee();
static void displayCount();
};
int employee :: count = 0;
employee :: employee()
{
string srushti,pune,sales;
count++;
cout << "Default Constructor Called";
```

```
name = srushti;
emp_id = 273;
dept=sales;
date_of_Join = 10/06/2024;
address = pune;
telephone = 2345678901;
}
void employee :: title()
{
cout << "\n- - -  Employee Management System- - - \n";
}
employee :: employee(string n, long int id, string dep, string doj, string add, int tele)
{
cout << "\nParameterized Constructor Called";
name = n;
emp_id = id;
dept = dep;
date_of_Join = doj;
address = add;
telephone = tele;
count++;
}
void employee :: display()
{
cout << "\nEmployee Name: " << name;
cout << "\nEmployee ID : " << emp_id;
cout << "\nDepartment of the employee: " << dept;
cout << "\nDate of Joining : " << date_of_Join;
cout << "\nAddress of employee: " << address;
cout << "\nTelephone no. of employee : " << telephone;
}
employee :: employee(employee &obj)
{
name = obj.name;
emp_id = obj.emp_id;
dept= obj.dept;
date_of_Join = obj.date_of_Join;
address = obj.address;
telephone = obj.telephone;
count++;
}
void employee :: display2()
{
cout << "\nCopy constructor Called";
```

```
}
employee :: ~employee()
{
cout << "\nDestructor is called!";
count--;
}
void employee :: displayCount()
{
cout << "\nCount : " << count;
}
int main()
{
employee obj1;
obj1.title();
string n;
long int id;
string dep;
string date;
string add;
int tele;
cout << “Name of the employee: ";
cin >> n;
cout << " employee ID: ";
cin >> id;
cout << " Department: ";
cin >> dep;
cout << " date of joining: ";
cin >> doj;
cout << " Address of the employee: ";
cin >> add;
cout << " telephone no. of the employee: ";
cin >> tele;
employee obj2(n,id,dep,date,add,tele);
obj2.display();
employee obj3 = obj2;
obj3.display2();
employee *ptr = new employee(n,id,dep,doj,add,tele); //Pointer
ptr -> display();
employee :: displayCount();
delete ptr;
return 0;
}
```

OUTPUT



```
Default Constructor Called
----Employee Management System----
 Name of the employee: srushti
 employee ID: 1079
 Department: sales
 date of joining: 01-04-2024
 Address of the employee: pune
 telephone no. of the employee: 2143658790

Parameterized Constructor Called
Employee Name: srushti
Employee ID : 1079
Department of the employee: sales
Date of Joining : 01-04-2024
Address of employee: pune
Telephone no. of employee : 2143658790

Copy constructor Called
Parameterized Constructor Called
Employee Name: srushti
Employee ID : 1079
Department of the employee: sales
Date of Joining : 01-04-2024
Address of employee: pune
Telephone no. of employee : 2143658790

Count : 4
Destructor is called!
Destructor is called!
Destructor is called!
Destructor is called!
```