**EMPLOYE CLASS**

**IMPLEMENTATION**

**LAB #** **04**

**Fall 2022**

**CSE-208**

**Object Oriented Programming**

Submitted by: **AIMAL KHAN**

Registration No.: **21PWCSE1996**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Dr. Nasru-min-Allah**

January 19, 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

and Technology, Peshawar

**Lab no. 4: Employ Class**

**Create a class called employee. This class maintains information about name (char\*), department (char\*), salary (double), and period of service in years (double).**

1. Provide a **no-argument constructor** to initialize the data members to some fixed values.
2. Provide a **4-argument constructor** to initialize the members to values sent from calling function.

(You have to make dynamic allocation for both name and department data members in constructor.)

1. Provide a **copy-constructor** that performs the deep copy of the data members.
2. Provide separate getter and setter functions for each attribute.
3. Provide an **input** function that takes all the values from user during run-time.
4. Provide a **display** function that displays all the information about a specific student to user.
5. Provide a **destructor** to free the memory allocated to name and department in constructor.

Write a driver program to test the functionality of the above-mentioned class.

**CODE:**

#include <iostream>

#include<string>

#include<cstring>

using namespace std;

class Employ{

private:

char\* name;

char\* department;

double salary;

double service;

public:

Employ():name(" "),department(" "), salary(0.00), service(0.00) {}

Employ(char na [], char de [], double sa, double se):name(na), department(de), salary(sa), service(se) {}

Employ(const Employ &e){

cout << "\nIn Custom Copy Constructor [Deep]" << endl;

int nameLength = strlen(e.name);

int departmentLength = strlen(e.department);

name = new char [nameLength + 1]; // '/0'

department = new char [departmentLength + 1];

strcpy(name, e.name);

strcpy(department, e.department);

salary = e.salary;

service = e.service;

}

void showEmploy ()const {

cout<<"Employ Data"<<endl;

cout<<"Name:\t"<<name<<"\tAddress: "<<(void \*)name<<endl;

cout<<"Department:\t"<<department<<"\tAddress: "<<(void \*)department<<endl;

cout<<"Salary:\t"<<salary<<endl;

cout<<"Service of years:\t"<<service<<endl<<endl;

}

void setEmploy(){

int nameLength = strlen(name);

name = new char [nameLength + 1];

cout<<"Enter Employ's name:\t";

gets(name);

int departmentLength = strlen(department);

department = new char [departmentLength + 1];

cout<<"Enter Employ's Department:\t";

gets(department);

cout<<"Enter Employ's Salary:\t";

cin>>salary;

cout<<"Enter Employ's Service:\t";

cin>>service;

}

~ Employ(){

cout<<"Object destroyed!";

delete [] name;

delete [] department;

}

};

int main(){

Employ e1("Aimal Khan", "DCSE", 49878.34, 21.2);

//Employ e1;

e1.showEmploy();

e1.setEmploy();

e1.showEmploy();

Employ e2 (e1);

e2.showEmploy();

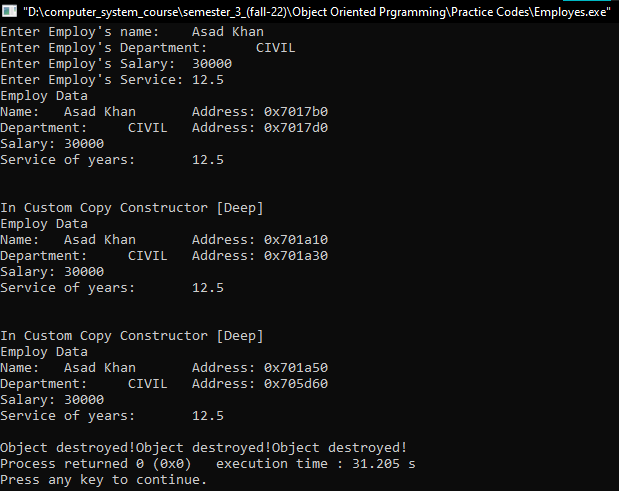
Employ e3 = e2;

e3.showEmploy();

return 0;

}

**Output:**

****