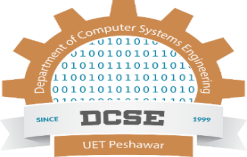
****

**University of engineering & technology Peshawar**

**Object oriented programming-lab**

**Lab report no#04**

**Spring 2020**

**Submitted by: Ashfaq Ahmad**

**Section: B**

**Reg No: 19PWCSE1795**

**Semester: 3rd**

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

Student signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Submitted to:**

**Eng: summayyea salahuddin**

**Department Of Computer System Engineering**

**Objectives of the Lab:**

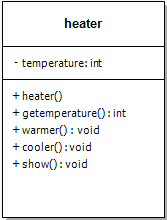
Objectives of the lab are to:

# Understand and implement parameterless and parameterized constructor in a class.

# Write a class (C++/Java) with overloaded constructors.

# Write a test program to use default copy constructor (C++).

* Understand the difference between a Shallow Copy and a Deep Copy.
* Understand the concept of dynamic memory allocation.
* Implement deep and shallow copy in a class (C++/Java).
* Use and test deep and shallow copy in a class.
* Understand and implement destructor in a class (C++/Python).

**Algorithm:**

**Employee**

First make class Employee

* Declare name, department, salary and period as private data members.

-salary,period:double

-name,department

* Define no argument constructor and also a parameterized constructor.

+Employee()

+input(): void

+show(): void

+~Employee()

+Employee(const Employee &obj)

* Define Show method to show the record of employee.
* Define Input method to input the record of employee.
* Define a copy constructor.
* Define Destructor.
* In main function, make objects of Employee to demonstrate the use of Employee.
* Call each function one after the other and display the show function as shown in the flow chart.

**Task no 1:**

C++: 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 parameter less constructor to initialize the data members to some fixed values.
2. Provide a 4-argument parameterized 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 an input function that takes all the values from user during run-time.
3. Provide a show function that shows all the information about a specific employee to user.
4. Provide a destructor to free the memory allocated to name and department in constructor.

* Write all the member function outside a C++ class. Write a driver program to test the functionality of the above-mentioned class.

**Flowchart:**

Show name,department,salary and period

Void Show()

Void Input()

End

End

Input name,department,salary and period

Start

End

Employee(char n[],char d[],int s,int p)

Employee()

Start

Start

Private: name,department,salary,period

End

Start

End

Name=n,department=d

Salary=s,period=p

Name,department🡨” ”

Salary,period🡨0

Start

Source code:

#include <iostream>

#include<string>

#include<cstring>

using namespace std;

class employee

{

private:

char \*name,\*department; //dynamic veriable

double salary,service;

public:

employee() // default constructor initialization

{

name=" ";

department=" ";

salary=0;

service=0;

}

employee(char n[],char d[], double s, double p) //perameterized constructor

{

name=n;

department=d;

salary=s;

service=p;

}

employee(const employee &e) // custom copy constructor - Deep

{

int len1 = strlen(e.name); // step 1: find length of input array

name = new char[len1+1]; // step 2: create name of length n + 1 ('/0')

int len2 = strlen(e.department);

department= new char [len2+1];

strcpy (name,e.name); // step 3: copy using strcpy

strcpy (department,e.department);

salary = e.salary;

service = e.service;

}

~employee()

{

cout << "Terminating object." << endl;

delete name;

delete department;

}

void input()

{

char n[100],d[100];

cout<<"please enter employer Data.....\n";

cout<<"Name: ";

cin>>n; //this cin for writing last name after spacing

int l1=strlen(n);

name=new char[l1+1];

strcpy(name,n);

cout<<"Department: ";

cin>>d;

int l2=strlen(d);

department= new char[l2+1];

strcpy(department,d);

cout<<"Salary: ";

cin>>salary;

cout<<"Period of service: ";

cin>>service;

}

void show(){

cout<<"\n\nemployee Data......."<<endl;

cout<<"Name: "<<name<<endl;

cout<<"department: "<<department<<endl;

cout<<"salary: "<<salary<<endl;

cout<<"preiod of service: "<<service<<"years"<<endl;

}

};

int main()

{

employee e;

e.input();

e.show();

employee e1(e); //function notation for copying object e in e1.

cout<<"\n\nAfter copying pre-existing object's data in newly created object by using function notation:\n";

e1.show();

employee e2=e; //assignment notation for calling copy constructor e.(we can also copying e data from e1)

cout<<"\n\nAfter copying pre-existing object's data in newly created object by using assignment notation: \n";

e2.show();

cout<<endl;

return 0;

};

Compilation:

