**Deep and Shallow Copy**

**LAB #** **04**

**Fall 2019**

**CSE208L Object Oriented Programming Lab**

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

“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:

**Engr. Sumayyea Salahuddin**

November 20, 2019

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**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).

# Activity # 01

**Title:**

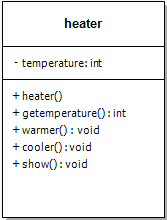
Make a class for Employee and model it using name, department, salary and period.

**Problem analysis:**

Create a class, **Employee** that contains name, department, salary and period. Define a constructor that takes no parameters. Define the methods **input** and **show** that inputs and shows the employee record respectively. Define a destructor and also a copy constructor. Demonstrate the use of Complex class.

**Algorithm:**

UML diagram for the above problem is given below:

* First make class Employee

**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 theeuse of Employee.
* Call each function one after the other and display the show function as shown in the flow chart.

**Flowchart:**

Void Show()

Void Input()

End

End

Show name,department,salary and period

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

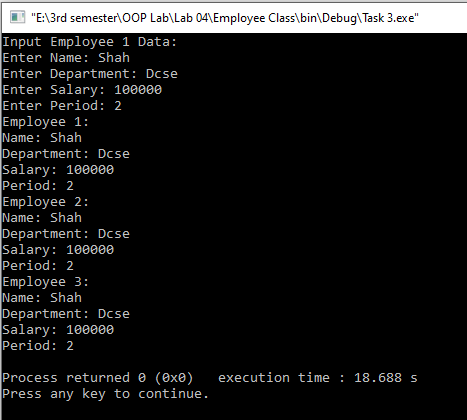
Name,department🡨” ”

Salary,period🡨0

Start

**In C++**

**Source code: Output:**

#include <iostream>

#include<string>

#include<cstring>

using namespace std;

class employee

{

char \*name;

char \*department;

double salary;

double period;

public:

employee()

{

name=" ";

department=" ";

salary=0;

period=0;

}

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

{

name=n;

department=d;

salary=s;

period=p;

}

employee (const employee &d)

{

int len=strlen(d.name);

name= new char[len+1];

int len2=strlen(d.department);

department= new char[len2+1];

strcpy(name,d.name);

strcpy(department,d.department);

salary=d.salary;

period=d.period;

}

~employee()

{

delete name;

delete department;

}

void input()

{

char n[100],d[100];

cout<<"Enter Name: ";

cin>>n;

int l1=strlen(n);

name=new char[l1+1];

strcpy(name,n);

cout<<"Enter Department: ";

cin>>d;

int l2=strlen(d);

department= new char[l2+1];

strcpy(department,d);

cout<<"Enter Salary: ";

cin>>salary;

cout<<"Enter Period: ";

cin>>period;

}

void show()

{

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

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

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

cout<<"Period: "<<period<<endl;

}

};

int main()

{

employee e1;

cout<<"Input Employee 1 Data:\n";

e1.input();

cout<<"Employee 1:\n";

e1.show();

employee e2=e1;

cout<<"Employee 2:\n";

e2.show();

employee e3(e1);

cout<<"Employee 3:\n";

e3.show();

return 0;

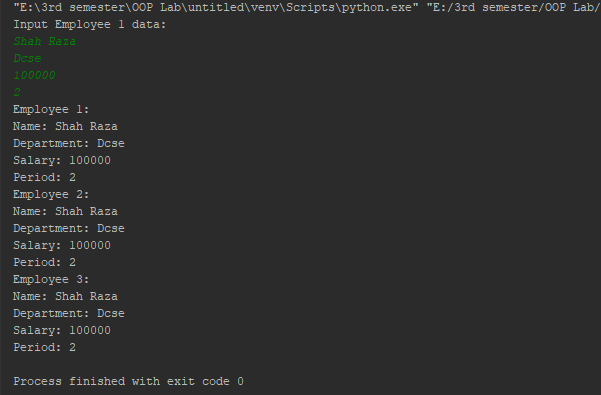
}

**In Python**

**Source code:**

import copy  
class employee:  
 def \_\_init\_\_(self,n="",d="",s=0,p=0):  
 self.name=n  
 self.department=d  
 self.salary=s  
 self.period=p  
 def datain(self):  
 self.name=input()  
 self.department=input()  
 self.salary=input()  
 self.period=input()  
 def show(self):  
 print("Name: " + self.name)  
 print("Department: " + self.department)  
 print("Salary: " + str(self.salary))  
 print("Period: " + str(self.period))  
  
e1= employee()  
print("Input Employee 1 data: ")  
e1.datain()  
print("Employee 1: ")  
e1.show()  
e2=copy.copy(e1)  
print("Employee 2: ")  
e2.show()  
e3=copy.deepcopy(e1)  
print("Employee 3: ")  
e3.show()

**Output:**



**Conclusion:**

This program helps us in understanding the basic concepts of Dynamic memory allocation, Destructor and Copy Constructor in different languages. It acts as a base for us and helps us in preparing ourselves for the higher level of programming. We get to know about the constructor and method in OOP with the help of this program.