# Third Year B. Tech (EL & CE)

**Semester: V Subject:** Object-Oriented Programming Lab

**Name: Shreerang Mhatre Class: TY**

**Roll No: 52 Batch: A3**

# Experiment No: 05

**Name of the Experiment: Virtual Function**

**Performed on: 22/11/2023**

**Submitted on: 22/11/2023**

**Problem Statement:**

Write a C++ program with base classEmployee and derive classes Class1\_Employee,

Class2\_Employee and Class3\_Employee.

Salary of an employee is calculated as per his/her designation.

Declare calculate salary () as a pure virtual function in the base class and

define it in respective derive classes to calculate salary of an employee.

**Output:**

D:\Object Oriented Programming\exp5>cd "d:\Object Oriented Programming\exp5\" && g++ virtual\_function.c++ -o virtual\_function && "d:\Object Oriented Programming\exp5\"virtual\_function  
Class1\_Employee  
Salary: $50000  
Class2\_Employee  
Salary: $60000  
Class3\_Employee  
Salary: $70000

**Code:**

#include <iostream>

// Base class

class Employee {

public:

virtual double calculateSalary() const = 0; // virtual function

virtual void displayType() const {

std::cout << "Base Employee" << std::endl;

}

};

// Derived class 1

class Class1\_Employee : public Employee {

public:

double calculateSalary() const override {

// Implement salary calculation logic for Class1\_Employee

return 50000.0;

}

void displayType() const override {

std::cout << "Class1\_Employee" << std::endl;

}

};

// Derived class 2

class Class2\_Employee : public Employee {

public:

double calculateSalary() const override {

// Implement salary calculation logic for Class2\_Employee

return 60000.0;

}

void displayType() const override {

std::cout << "Class2\_Employee" << std::endl;

}

};

// Derived class 3

class Class3\_Employee : public Employee {

public:

double calculateSalary() const override {

// Implement salary calculation logic for Class3\_Employee

return 70000.0;

}

void displayType() const override {

std::cout << "Class3\_Employee" << std::endl;

}

};

int main() {

Class1\_Employee employee1;

Class2\_Employee employee2;

Class3\_Employee employee3;

// Displaying employee types and their salaries

employee1.displayType();

std::cout << "Salary: $" << employee1.calculateSalary() << std::endl;

employee2.displayType();

std::cout << "Salary: $" << employee2.calculateSalary() << std::endl;

employee3.displayType();

std::cout << "Salary: $" << employee3.calculateSalary() << std::endl;

return 0;

}





