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;
}
```

virtual_function.c++

```cpp
/*

Shreerang Mhatre
Rollno - 52
Batch - A3
Expno - 5

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.

*/

#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;
```

```
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

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

D:\Object Oriented Programming\exp5>
```

virtual_function.c++

```cpp
// 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;
    }
};
```

```
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

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

D:\Object Oriented Programming\exp5>
```

```cpp
        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;
}
```

```
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

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

D:\Object Oriented Programming\exp5>
```