

1. Write a program in C++ to implement single inheritance. Derive class typist class from class staff. Public members of class staff such as staff::getdata() and staff::display() are inherited to class typist. Access the member functions using objects of derived class.

C++ Code:

```
#include <iostream>
using namespace std;

// Base class
class Staff {
    struct Data
    {
        int pid;
        char name[100];
        float salary;
    } ajay;

public:
    void getdata()
    {
        ajay = {221077, "Ajay", 100000.00};
    }
    void display()
    {
        cout << "PID no: " << ajay.pid << endl;
        cout << "Name: " << ajay.name << endl;
        cout << "Salary: " << ajay.salary << endl;
    }
};

class Typist : public Staff {
public:
    void print()
    {
        Staff staff;
        staff.getdata();
        staff.display();
    }
};

int main() {
    Typist typer;
    typer.print();
    return 0;
}
```

Output

```

  TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE  SERIAL MONITOR  COMMENTS

PS C:\Users\Ajay kumar\Desktop\SEIT\PCPF\Lab\Inheritance using Java & C++\C++> g++

PID no: 221077
Name: Ajay
Salary: 100000

PS C:\Users\Ajay kumar\Desktop\SEIT\PCPF\Lab\Inheritance using Java & C++\C++> 
```

2. Write a program in C++ to implement multiple inheritance. Create class Rectangle having two parent classes Area and Perimeter. Class 'Area' has a function getArea(int l, int b) which returns area. Class 'Perimeter' has a function getPerimeter (int l, int b) which returns the perimeter. Access the member functions using objects of derived class.

C++ Code;

```
#include <iostream>
using namespace std;

// Area class with a function to calculate area
class Area {
public:
    int getArea(int l, int b)
    {
        return l * b;
    }
};

// Perimeter class with a function to calculate perimeter
class Perimeter {
public:
    int getPerimeter(int l, int b)
    {
        return 2 * (l + b);
    }
};

// Rectangle class inheriting from Area and Perimeter
class Rectangle : public Area, public Perimeter
{
public:
};

int main() {
    int length, breadth;
    cout << "Enter the length of the rectangle: ";
    cin >> length;
    cout << "Enter the breadth of the rectangle: ";
    cin >> breadth;

    Rectangle rect;

    int area = rect.getArea(length, breadth);
    int perimeter = rect.getPerimeter(length, breadth);
    cout << "Area of the rectangle: " << area << endl;
    cout << "Perimeter of the rectangle: " << perimeter << endl;
    return 0;
}
```

Output:

TERMINAL

PROBLEMS

OUTPUT

DEBUG CONSOLE

SERIAL MONITOR

COMMENTS

```
PS C:\Users\Ajay kumar\Desktop\SEIT\PCPF\Lab\Inheritance using Java & C++\C++> g++ inheritance1.cpp -o inheritance1 ; if ($?) { g++ inheritance1.cpp -o inheritance1 } ; if ($?) {
```

```
Enter the length of the rectangle: 4
```

```
Enter the breadth of the rectangle: 3
```

Area of the rectangle: 12

Perimeter of the rectangle: 14

```
PS C:\Users\Ajay kumar\Desktop\SEIT\PCPF\Lab\Inheritance using Java & C++\C
```

3. Write a program in Java to implement hierarchical inheritance. Create PermanentEmp class and TemporaryEmp class as the subclasses. Create class Employee as the super class that defines the salary of an employee. Class PermanentEmp has a hike of 35% of the amount defined in class Employee and Class TemporaryEmp has a hike of 20% of the amount defined in class Employee. Access the member functions using objects of derived class.

Java Code:

```
class Employee {
    protected double salary;

    public Employee(double salary) {
        this.salary = salary;
    }

    public double getSalary() {
        return salary;
    }
}

class PermanentEmp extends Employee {
    public PermanentEmp(double salary) {
        super(salary);
    }

    public double getHike() {
        return salary * 0.35; // 35% hike
    }
}

class TemporaryEmp extends Employee {
    public TemporaryEmp(double salary) {
        super(salary);
    }

    public double getHike() {
        return salary * 0.20; // 20% hike
    }
}

public class Main {
    public static void main(String[] args) {
        // Create PermanentEmp object
        PermanentEmp permanentEmployee = new PermanentEmp(50000);

        // Create TemporaryEmp object
        TemporaryEmp temporaryEmployee = new TemporaryEmp(30000);

        // Access the member functions using objects of derived classes
        double permanentSalary = permanentEmployee.getSalary();
```

```

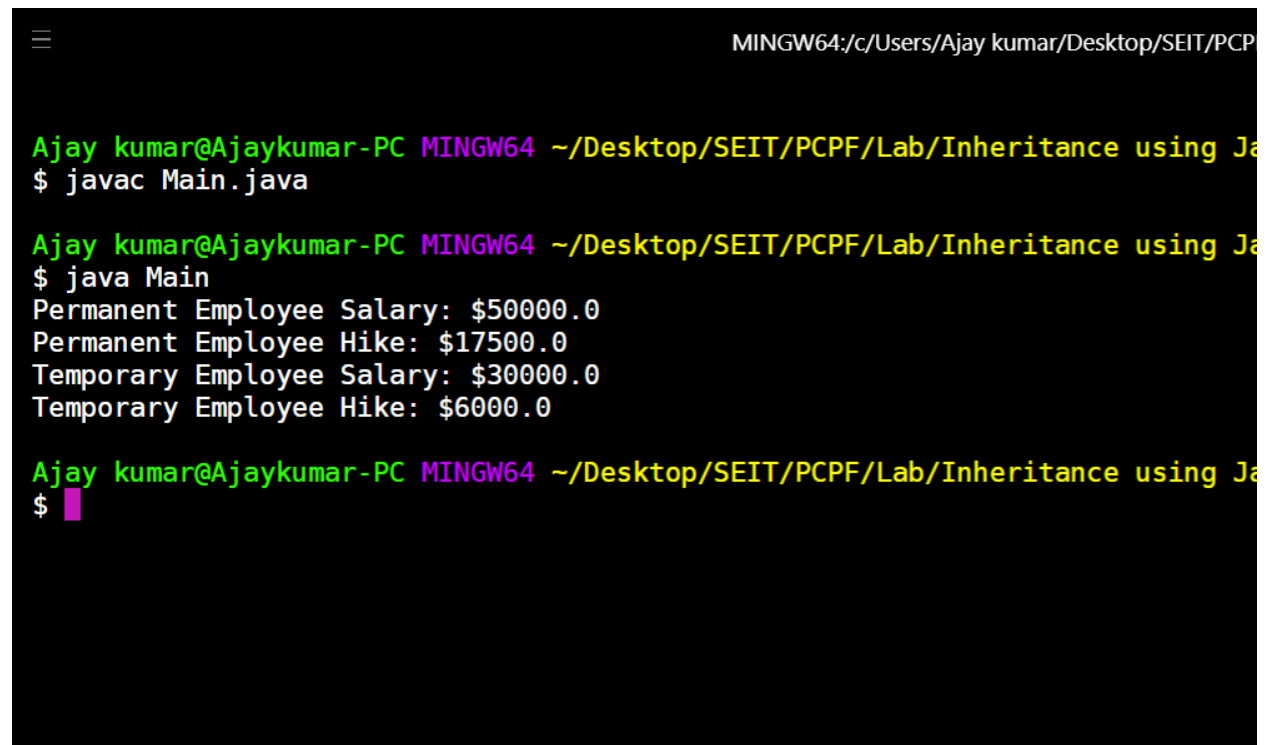
        double permanentHike = permanentEmployee.getHike();

        double temporarySalary = temporaryEmployee.getSalary();
        double temporaryHike = temporaryEmployee.getHike();

        System.out.println("Permanent Employee Salary: $" + permanentSalary);
        System.out.println("Permanent Employee Hike: $" + permanentHike);
        System.out.println("Temporary Employee Salary: $" + temporarySalary);
        System.out.println("Temporary Employee Hike: $" + temporaryHike);
    }
}

```

Output:



```

MINGW64:/c/Users/Ajay kumar/Desktop/SEIT/PCPF
Ajay kumar@Ajaykumar-PC MINGW64 ~/Desktop/SEIT/PCPF/Lab/Inheritance using Ja
$ javac Main.java

Ajay kumar@Ajaykumar-PC MINGW64 ~/Desktop/SEIT/PCPF/Lab/Inheritance using Ja
$ java Main
Permanent Employee Salary: $50000.0
Permanent Employee Hike: $17500.0
Temporary Employee Salary: $30000.0
Temporary Employee Hike: $6000.0

Ajay kumar@Ajaykumar-PC MINGW64 ~/Desktop/SEIT/PCPF/Lab/Inheritance using Ja
$ 

```