**Open Closed Principle**

**Problem:**

**public** **interface** CalculatorOperation {

**double** perform(char operator);

}

**public** **class** Calculation **implements** CalculatorOperation {

**private** **double** left;

**private** **double** right;

**private** **double** result;

public Calculation(double a,double b)

{

This.left=a;

This.right=b;

}

@Override

**public** **double** perform(char operator)

{

switch(operator)

{

Case ‘+’ ; result = left + right; break;

Case ‘-‘ : result = left - right; break;

Default : return 0.0;

}

**return** result;

}

}

**Public class DriverApp**

**{**

**Psvm()**

**{**

Calculation c1=new Calculation(10,20);

System.out.println(C1.perform(‘+’));

**}**

**}**

**Soultion:**

**public** **interface** CalculatorOperation {

**double** perform();

}

**public** **class** Addition **implements** CalculatorOperation {

**private** **double** left;

**private** **double** right;

**private** **double** result;

// constructor, getters and setters

@Override

**public** **double** perform() {

result = left + right;

**return** result;

}

**public** Addition() {}

**public** Addition(**double** left, **double** right) {

**this**.left = left;

**this**.right = right;

}

}

**public** **class** Calculator {

**public** **double** calculate(CalculatorOperation operation) {

**if** (operation == **null**) {

**throw** **new** InvalidParameterException("Cannot perform operation");

}

**return** operation.perform();

}

}

**public** **class** DriverApp {

**public** **static** **void** main(String[] args) {

Calculator c1=**new** Calculator();

Addition a1=new Addition(50,20);

System.***out***.println(c1.calculate(s1));

}

}