

TASK1:

Bloaters(Long Method)

1. God Class

-eki classs e sob kaj kora hoise

2. Bloaters(primitive obsession):

only double type a and b.

3.(Dispensables)Dead Code :

```
// never reach  
throw new Error();  
In both class
```

4. Extra Comment:

```
// never reach
```

TASK2:

Calculator.java er calculateBiImpl() and calculateMono() method e if-else chains ache
every operation er jonno sep condition check korte
new operation korte existing code modify korte hobe
Violated Principle: Open/Closed Principle (OCP)

Strategy Design Pattern;

Violated Principle: Open/Closed Principle (OCP)

it violates open closed principle that to add a

suppose i want to add a new operation like : minus and then *5
i have to modify the existing code

```
public enum MonoOperatorModes {  
    square, squareRoot, oneDividedBy, cos, sin, tan, log, rate, abs, ln, minusthenmultiply5  
}
```

```
if (mode.equals(BiOperatorModes.minusthenmultiply5)) {  
    return (num1 - num2 ) * 5;  
}
```

Violated Principle: (SRP)

solution:

create separate interface for the operations:

****Strategy Design Pattern and Factory DP:**

interface: OperationStrategy.java

Mono op classes: Square.java , SquareRoot.java , oneDividedBy.java , cosineOperation.java , sin.java, tan.java, log.java, rate.java, abs.java, ln.java

Bi: normalOp.java, add.java, minus.java, multiply.java, divide.java , xpowerofy.java

Calculator.java (main class)

TASK3:

Task 3:

Add a new class MorethanOneOperation ;