

COM 102 – OBJECT ORIENTED PROGRAMMING POSTLAB #8- KEY

2- Error Correction (5x4= 20 pts)

```
public interface Enable {  
    void on(){  
        };           → remove curly braces because it cannot be include a body.  
  
    void off();  
}  
public class Projector extends Enable { → There is no is-a relationship  
between Projector and Enable, That's why, use keyword implements instead  
of extends.  
  
    private boolean state;  
    private int lightDuration;  
  
    public Projector() {  
        state = false;  
    }  
  
    @Override  
    public void on() {  
        setState(true);  
    }  
  
    @Override  
    public void off() {  
        setState(false);  
    }  
  
    public boolean isState() {  
        return state;  
    }  
  
    public void setState(boolean state) {  
        this.state = state;  
    }  
  
    public int getLightDuration() {  
        return lightDuration;  
    }  
  
    public void setLightDuration(int lightDuration) {  
        if(lightDuration>=0){  
            this.lightDuration = lightDuration;  
        }  
        else{
```

```

        this.lightDuration = 0;
    }
}

```

public class SimpleAritmeticCalculator **implements** Enable throws
 IllegalArgumentException {
 → Remove throws IllegalArgumentException, because it does not throw
 such an exception.

```
private boolean state;
```

```
public SimpleAritmeticCalculator() {
    state = false;
}
```

```
@Override
public void on() {
    setState(true);
}
```

@Override
 public void off(); → you have to implement this method because your
 class implements Enable interface. That's why, add the following lines as
 the method body.

```
{
    setState(false);
}
```

```
public boolean isState() {
    return state;
}
```

```
public void setState(boolean state) {
    this.state = state;
}
```

```
public int add(int... numbers){
    int sum = 0;
    for(int number:numbers){
        sum += number;
    }
    return sum;
}
```

```
public int subtract(int... numbers){
    int difference = numbers[0];

    for(int i = 1; i<numbers.length;i++){
        difference -= numbers[i];
    }
    return difference;
}
```

```
public int multiply(int... numbers){
```

```

    int result = 1;

    for(int number:numbers){
        result *= number;
    }
    return result;
}

public int divide(int numerator, int denominator){
    int result = 0;

    try{
        result = numerator/denominator;

    }
    catch(DivideByZeroException arithmeticException){ → use
ArithmeticException type instead DivideByZeroException or create a
DivideByZeroException.
        System.err.printf( "\nException: %s\n", arithmeticException );
        System.out.println("Zero is an invalid denominator. Please try
again.\n" );

    }
    catch(Exception e){
        System.out.println("An Unknown Error has occured!!");
        e.printStackTrace();
    }

    return result;
}
}

```