Program 1 : Write a code which will contain 2 numbers. Use the following operators on it.
1. less than (<)
2. greater than (>)
3. less than or equal to (<=)
4. greater than or equal to (>=)
5. equivalent (==)
6. not equivalent (!=)

Solution :

```
class Program1{
	public static void main(String[] args){
		int a = 10;
		int b = 20;
		System.out.print("Operator 1(<) : ");
		System.out.print("a<b : " + (a<b) + "\n");
		System.out.print("Operator 2(>) : ");
		System.out.print("a<b : " + (a>b) + "\n");
		System.out.print("Operator 3(<=) : ");
		System.out.print("a<b : " + (a<=b) + "\n");
		System.out.print("Operator 4(>=) : ");
		System.out.print("a<b : " + (a>=b) + "\n");
		System.out.print("Operator 5(==) : ");
		System.out.print("a<b : " + (a==b) + "\n");
		System.out.print("Operator 6(!=) : ");
		System.out.print("a<b : " + (a!=b) + "\n");
	}
}
/*
sk@sk-Lenovo-ideapad-110-15ISK:~/Desktop/Shalaka/Solution$ java Program1
Operator 1(<) : a<b : true
Operator 2(>) : a<b : false
Operator 3(<=) : a<b : true
Operator 4(>=) : a<b : false
Operator 5(==) : a<b : false
Operator 6(!=) : a<b : true
sk@sk-Lenovo-ideapad-110-15ISK:~/Desktop/Shalaka/Solution$
*/
```

Program 2 : Write a code which will contain one number ‘num’. Use the following operators on it.
1. num >>>= 2
2. num <<= 3
3. num ^= 2
4. num = ++num
5.num = ~num

Solution :

```
class Program2{
	public static void main(String[] args){
		int num = 100;
		System.out.print("Operator 1(>>>) : ");
		System.out.print("num>>>= 2 : " + (num>>>= 2) + "\n");
		System.out.print("Operator 2(<<) : ");
```

```
                System.out.print("num<<= 3 : " + (num<<= 3) + "\n");
                System.out.print("Operator 3(^) : ");
                num = 5;
                System.out.print("num ^= 2 : " + (num ^= 2) + "\n");
                System.out.print("Operator 4(++) : ");
                System.out.print("num = ++num : " + (num = ++num) + "\n");
                System.out.print("Operator 5(~) : ");
                System.out.print("num = ~num : " + (num = ~num) + "\n");
        }
}
/*
sk@sk-Lenovo-ideapad-110-15ISK:~/Desktop/Shalaka/Solution$ javac Program2.java
sk@sk-Lenovo-ideapad-110-15ISK:~/Desktop/Shalaka/Solution$ java Program2
Operator 1(>>>) : num>>>= 2 : 25
Operator 2(<<) : num<<= 3 : 200
Operator 3(^) : num ^= 2 : 7
Operator 4(++) : num = ++num : 8
Operator 5(~) : num = ~num : -9
sk@sk-Lenovo-ideapad-110-15ISK:~/Desktop/Shalaka/Solution$
*/
```

Program 3 : What will will the output of the statement if i=5, j=4
(i++ >= j++) && (--j < i++)
(--i < j--) || (j++ >= ++i)

Solution :

```
class Program3{
        public static void main(String[] args){
                int i = 5;
                int j = 4;
                System.out.print("(i++ >= j++) && (--j < i++) : " + ((i++ >= j++) && (--j < i++))
+ "\n");
                System.out.print("(--i < j--) || (j++ >= ++i) : " + ((--i < j--) || (j++ >= ++i)) + "\
n");

        }
}
/*
sk@sk-Lenovo-ideapad-110-15ISK:~/Desktop/Shalaka/Solution$ javac Program3.java
sk@sk-Lenovo-ideapad-110-15ISK:~/Desktop/Shalaka/Solution$ java Program3
(i++ >= j++) && (--j < i++) : true
(--i < j--) || (j++ >= ++i) : false
sk@sk-Lenovo-ideapad-110-15ISK:~/Desktop/Shalaka/Solution$
*/
```

Program 4 : Write the binary number of the following decimal numbers (Do it on paper/notebook with proper explaination)
1.56, 2.34, 3.65, 4.103, 5.97

Solution :

Q.4 Write the binary number for following decimal numbers.

1] 56

56
- 32 ✓
24
- 16 ✓
8 ✓

. . . . . 128 64 32 16 8 4 2 1
0000 0000 0000 0000 0000 0000 0000 0000
2^{31} . . . . . $2^{15}\,2^{14}\,2^{13}\,2^{12}\,2^{11}\,2^{10}\,2^{9}\,2^{8}\,2^{7}\,2^{6}\,2^{5}\,2^{4}\,2^{3}\,2^{2}\,2^{1}\,2^{0}$
0000 0000 0000 0000 0000 0000 0011 1000

∴ 56 ⇒ **0011 1000**

2] 34

34
- 32 ✓
2 ✓

0000 0000 0000 0000 0000 0000 0000 0000
. . . . . 128 64 32 16 8 4 2 1
0000 0000 0000 0000 0000 0000 0010 0010

∴ **34 ⇒ 00100010**

3] 65

65
- 64 ✓
1 ✓

. . . . . 128 64 32 16 8 4 2 1
0000 0000 0000 0000 0000 0000 0000 0000
2^{31} . . . . . $2^{15}\,2^{14}\,2^{13}\,2^{12}\,2^{11}\,2^{10}\,2^{9}\,2^{8}\,2^{7}\,2^{6}\,2^{5}\,2^{4}\,2^{3}\,2^{2}\,2^{1}\,2^{0}$
0000 0000 0000 0000 0000 0000 0100 0001

∴ **65 ⇒ 0100 0001**

4] 103

① 103 − 64 ✓ = 39 ② 39 − 32 ✓ = 7 ③ 7 − 4 ✓ = 3 ④ 3 − 2 ✓ = 1 ⑤ 1

∴

. . . . . 128 64 32 16 8 4 2 1

0000 0000 0000 0000 0000 0000 0000 0000

2^{31} . . . . . $2^{15}\,2^{14}\,2^{13}\,2^{12}\,2^{11}\,2^{10}\,2^{9}\,2^{8}\,2^{7}\,2^{6}\,2^{5}\,2^{4}\,2^{3}\,2^{2}\,2^{1}\,2^{0}$

0000 0000 0000 0000 0000 0000 0110 0111

103 ⇒ 01100111

5] 97

① 97 − 64 ✓ = 33 ② 33 − 32 ✓ = 1 ③ 1

. . . . . 128 64 32 16 8 4 2 1

0000 0000 0000 0000 0000 0000 0000 0000

2^{31} . . . . . $2^{15}\,2^{14}\,2^{13}\,2^{12}\,2^{11}\,2^{10}\,2^{9}\,2^{8}\,2^{7}\,2^{6}\,2^{5}\,2^{4}\,2^{3}\,2^{2}\,2^{1}\,2^{0}$

0000 0000 0000 0000 0000 0000 0110 0001

∴ **97 ⇒ 01100001**

Program 5 : Write 2 classes as Pune, Mumbai. Add the following things :
class Pune : static variable : int covidActiveCases
instance variable : int totalCases
static method : noCorona
instance method : lockdown
noLockdown
class Mumbai : static variable : int covidActiveCases
instance variable : int totalCases
static method : noCorona
instance method : lockdown
noLockdown
Increment both the variables in non static method. Call non static methods twice, display the output and observe the changes.
Draw JVM architecture for the same.

Solution :

```
class Pune{
	static int activeCases = 23000;
	int totalCases = 28000;

	static void noCorona(){
		System.out.println("Free Pune..No CORONA, No Tension !!");
	}
	public void lockdown(){
		System.out.println("\nLockdown : ");
		activeCases++;
		totalCases++;
		System.out.println("Active Cases : " + activeCases);
		System.out.println("Total Cases : " + totalCases);
	}
	public void noLockdown(){
		System.out.println("\nNo Lockdown : ");
		activeCases++;
		totalCases++;
		System.out.println("Active Cases : " + activeCases);
		System.out.println("Total Cases : " + totalCases);
	}

}
class Mumbai{
	static int activeCases = 21000;
	int totalCases = 84000;

	static void noCorona(){
		System.out.println("Free Mumbai...No CORONA, No Tension !!");
	}
	public void lockdown(){
		System.out.println("\nLockdown : ");
```

```
                activeCases++;
                totalCases++;
                System.out.println("Active Cases : " + activeCases);
                System.out.println("Total Cases : " + totalCases);
        }
        public void noLockdown(){
                System.out.println("\nNo Lockdown : ");
                activeCases++;
                totalCases++;
                System.out.println("Active Cases : " + activeCases);
                System.out.println("Total Cases : " + totalCases);
        }
}
class Program5{
        public static void main(String[] args){
                System.out.println("\nPUNE : \n");
                Pune p1 = new Pune();
                p1.noCorona();
                p1.lockdown();
                p1.noLockdown();

                Pune p2 = new Pune();
                p2.lockdown();
                p2.noLockdown();

                System.out.println("\nMUMBAI : \n");
                Mumbai m1 = new Mumbai();
                m1.noCorona();
                m1.lockdown();
                m1.noLockdown();
                Mumbai m2 = new Mumbai();
                m2.lockdown();
                m2.noLockdown();
        }
}
/*
sk@sk-Lenovo-ideapad-110-15ISK:~/Desktop/Shalaka/Solution$ java Program5

PUNE :

Free Pune..No CORONA, No Tension !!

Lockdown :
Active Cases : 23001
Total Cases : 28001

No Lockdown :
Active Cases : 23002
Total Cases : 28002

Lockdown :
Active Cases : 23003
```

Total Cases : 28001

No Lockdown :
Active Cases : 23004
Total Cases : 28002

MUMBAI :

Free Mumbai...No CORONA, No Tension !!

Lockdown :
Active Cases : 21001
Total Cases : 84001

No Lockdown :
Active Cases : 21002
Total Cases : 84002

Lockdown :
Active Cases : 21003
Total Cases : 84001

No Lockdown :
Active Cases : 21004
Total Cases : 84002
sk@sk-Lenovo-ideapad-110-15ISK:~/Desktop/Shalaka/Solution$
***/**