1. Problem statement: Write a program to print the result of the following expressions provided the integer variable

a is 20 and b is 10.

int b = a-- - --a;

int c = a--;

int d = a>>2;

int e = a&b;

2. code with comments:

import java.lang.*;

public class AddOp {

        public static void main(String[] args) {

                        int a = 20, b=10;

                        b = a-- - --a;   // here b= 20 - 18

                         System.out.println(b); // Because post decrement 1 and pre-decrement 1 makes -2 from 20

                        int c = a--; // value of a is already 18 from the above execution and there is no operation after post

                         System.out.println(c);

                        int d = a>>2; // Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.

                        System.out.println(d);

                        int e = a&b; // Binary AND Operator copies a bit to the result if it exists in both operands.

```
        System.out.println(e);




    }



}
```

3. Explanation of the code:

Here b= 20 and a =18. So for the first output statement b=20-18, Because post decrement 1 and pre-decrement 1 makes -2 from 20. Now when executing the second output statement, value of a is already 18 from the above execution and there is no operation after post . Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand. Binary AND Operator copies a bit to the result if it exists in both operands.

4. Result flow in detail:

Here b= 20 and a =18. So for the first output statement b=20-18, Because post decrement 1 and pre-decrement 1 makes -2 from 20. Now when executing the second output statement, value of a is already 18 from the above execution and there is no operation after post . Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand. Binary AND Operator copies a bit to the result if it exists in both operands.

5. Output screenshot:

```
          System.out.println(AddPrg(15, 32));
                           ^

  symbol:    method AddPrg(int,int)
  location: class AddPrg
1 error

C:\Training_Assignment>javac AddPrg.java

C:\Training_Assignment>java AddPrg
47

C:\Training_Assignment>
C:\Training_Assignment>
C:\Training_Assignment>
C:\Training_Assignment>
C:\Training_Assignment>
C:\Training_Assignment>javac AddOp.java

C:\Training_Assignment>java AddOp
2
18
4
0

C:\Training_Assignment>
```