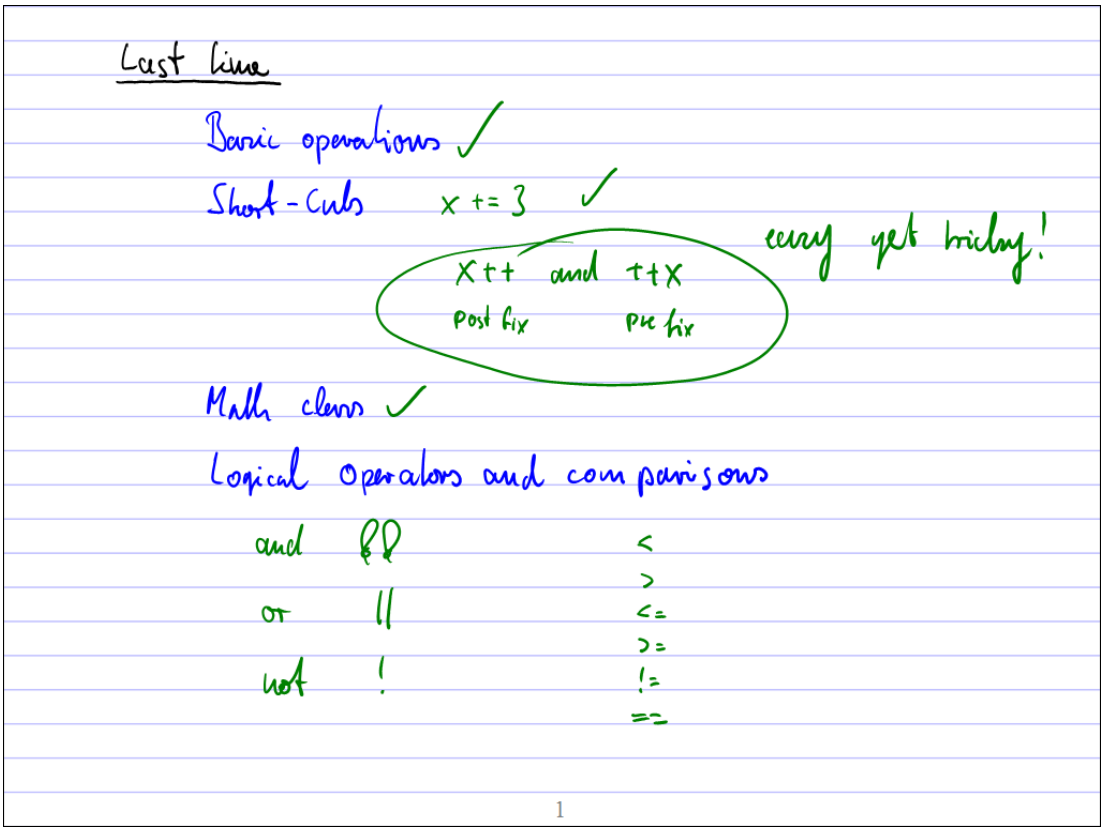
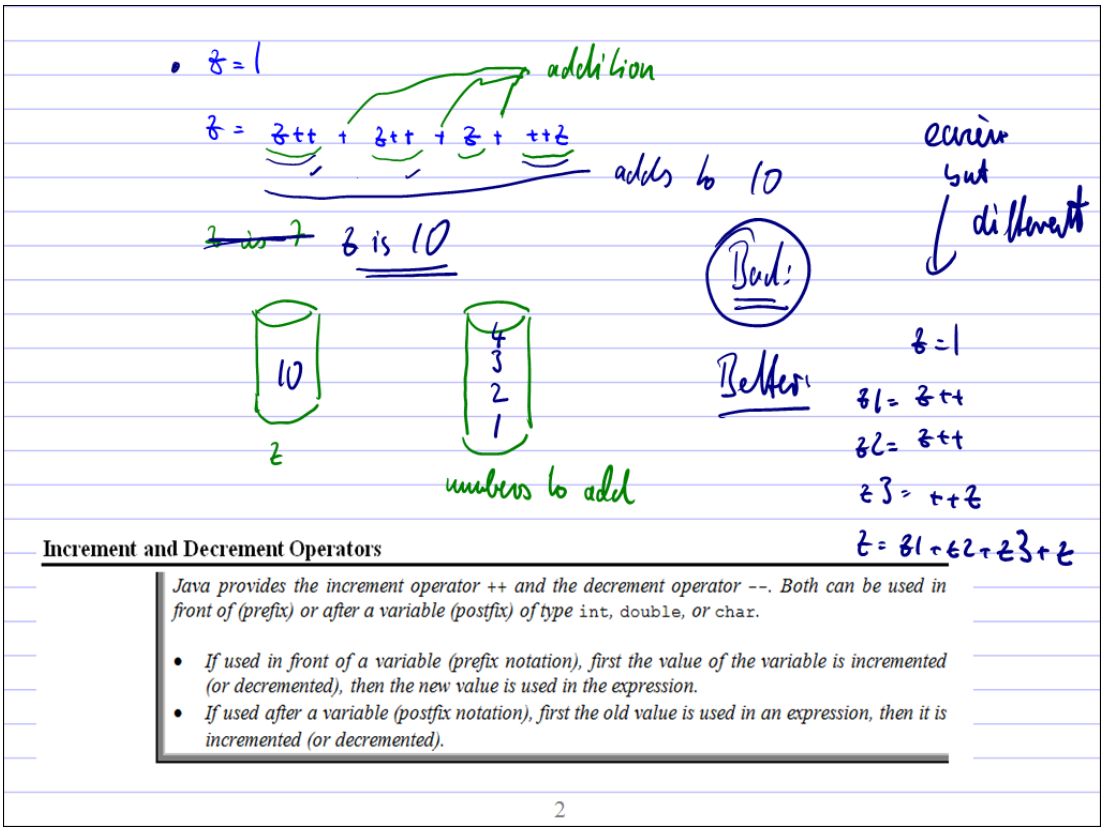
Panel 1



Panel 2



Panel 3

1. Determine if the following statements are `true` or `false`, using:

```
int x = 5, y = 9, z = 2;
boolean r = false, result;
```

Don't forget rest of Question 2

a. `result = ( y < z );` false

9 2

b. `result = !( ( r || ( y < x ) ) && ( 5 >= x ) );`

!((false || false) && true)

!(false && true)

!false

true

c. `result = (!r && ( (--y) > (z + x)(2) ));`

false

19 9 5 2 1

d. `result = (((21 - z) == (((y++) + x + z) + 1)) && (!r == true));`

10

3

Panel 4

Conditional Execution

The Simple `if` Statement

A simple `if` statement specifies the condition under which a certain block of code executes, using the syntax

```
if (condition)
   blockOfCode;
```

where `condition` is an expression of `boolean` type, such as a test. If `condition` evaluates to `true` the `blockOfCode` executes, otherwise execution will skip that block. In either case, execution resumes immediately after the `blockOfCode`.

Ex: z has an unknown value. Write code to print "positive" if z is positive!

optional if only one line.

```
if (z > 0)
{
    System.out.println("positive");
}
```

A code block is either a single line or code enclosed in { ... }

4

Panel 5

Consider the following program code (where x has previously been declared as a `double` variable of unknown value):

```
if (x == 0.0)
   System.out.println("x is zero");
   System.out.println("Can not divide by x");
```

← code block!

What is displayed if `x` is equal to `0.0`? How about if `x = 10.0`? Is this code what was (most likely) intended? If not, fix the code accordingly.

Say x = 0 : "x is zero"
Can not divide by zero

x = 10.0 : "Can not divide by x

5

Panel 6

The `if-else` Statement with Alternative

An `if-else` statement with alternative lets you use a `boolean` condition, usually a test, to determine which of two blocks of code executes, using the syntax:

```
if (condition)
   blockOfCode1;
else
   blockOfCode2;
```

If `condition` returns `true`, the `blockOfCode1` executes, otherwise `blockOfCode2` executes. After either of the two blocks has executed, the lines immediately after the `else` block continue to execute.

The Nested `if-else-if` Statement

A nested `if-else-if` statement lets you choose which of n blocks of code plus one optional block of code executes, based on `boolean` conditions or tests. The syntax is:

```
if (condition1)
   codeBlock1;
else if (condition2)
   codeBlock2;
...
else if (conditionN)
   codeBlockN;
[ else
   optionalBlock;]
```

If `condition1` evaluates to `true` execute `codeBlock1`, otherwise check `condition2`. If `condition2` evaluates to `true` execute `codeBlock2`, otherwise check `condition3`. Continue in this fashion. If `conditionN` evaluates to `true`, execute `codeBlockN`, otherwise execute `optionalBlock`, if defined, or continue with regular execution.

The k^{th} block of code executes if all conditions prior to the k^{th} one evaluate to `false` and the k^{th} one is `true`. The `optionalBlock` of code executes if all conditions evaluate to `false`.

6

Panel 7

If x is a `double` whose value is not known, write some code that will print out "`x is positive`" or "`x is not positive`", depending on the value of x.

```
public static void main(String args[])
  {
     double x = -60.0;

     if (x > 0.0)
     {
        System.out.println("x is positive");
     }
     else
     {
        System.out.println("x is NOT positive");
     }
  }
```

Suppose a `double` variable `score` contains a number between `0` and `100`, reflecting a student's performance on a test. Based on that `score` write a code segment that displays the corresponding letter grade.

100 - 90 = A
89 - 80 B
79 - 70 C
69 - 60 D
less F

7

Panel 8

if ((x >= 90) && (x <= 100))
~~blah~~ ;

is legal

```
public static void main(String args[])
{
   double score = -60;

   if (score > 100)
      System.out.println("Invalid");
   else if (score >= 90)
      System.out.println("A");
   else if (score >= 80)
      System.out.println("B");
   else if (score >= 70)
      System.out.println("C");
   else if (score >= 60)
      System.out.println("D");
   else if (score >= 0)
      System.out.println("F");
   else
      System.out.println("Invalid");
}
```

slickest solution,
but others are
correct, too!

8