Panel 1

**Scope:** Where a variable is defined is valid inside segment where it is defined. If defined twice, closer one counts.

**Field:** a variable at class level, accessible to all methods.

```java
public class Test {
    double x = 10; // field
    public void change (double x, double y) {
        x = 20;  y = 30;
    }
}
```

Panel 2

A method returns at most one value!
If methods need to modify 2 or more variables = use fields.

Ex: Compute compound interest, 6.5% per year, for $10000 over 10 years.
Print out interest + new amount per year.

<table>
<thead>
<tr>
<th>YEAR</th>
<th>Interest</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>$10000</td>
</tr>
<tr>
<td>1</td>
<td>$650</td>
<td>$10650</td>
</tr>
<tr>
<td>2</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Nice try for marketed if poss.
Panel 3

7) One method for interest
   One method for new amount

8) One method to change interest + amount!
   \[ \text{requires fields!} \]

Note: w/o compounding: \[ \$10000 \text{ at } 5\% \text{ for } 10 \text{ year} \]
   \[ = \$16500 \]

   w/ compounding: \[ \$18771 \]

Panel 4

Option: Ask for periods of compounding
per year, then print amounts!

I will explain \underline{Decimal} Formal on handout

Next Assignment: Secret Code

\[ \Rightarrow \text{BERT } \leftrightarrow \text{ CPSU} \]