Encapsulation, Test-Driven Development

1) variables can hold primitives
variables that hold object references

```
<table>
<thead>
<tr>
<th>a</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>4.5</td>
</tr>
<tr>
<td>c</td>
<td>true</td>
</tr>
</tbody>
</table>
```

```
String t = s
```

```
? t
```

```
Assign reference vars => sharing
```

execution stack
where methods run

object heap
Encapsulation — each object protects its data

- declare instance variables private or protected
  - enforces encapsulation
- define methods to limit access to internal rep.
  - don't want to "expose the internal rep"
Test-Driven Development

Game:

Test writer

Write a test

Developer

Implement simplest thing possible to pass the test

add to the tests
to force a more accurate implementation

Point:

- The tests fully capture the specification
- Anything that passes the tests is acceptable