**Relation operators:**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a&lt;b</td>
<td>a is less than b</td>
</tr>
<tr>
<td>a&gt;b</td>
<td>a is greater than b</td>
</tr>
<tr>
<td>a&lt;=b</td>
<td>a is less than or equal to b</td>
</tr>
<tr>
<td>a&gt;=b</td>
<td>a is greater than or equal to b</td>
</tr>
<tr>
<td>a==b</td>
<td>a equals b</td>
</tr>
<tr>
<td>a!=b</td>
<td>a does not equal b</td>
</tr>
</tbody>
</table>

**Two string**

- use `.equals()`

```
Ex: String answer = "Yes";
boolean compare = answer.equals("yes");
```

**Logical operators**

- `a||b` logical or
- `a&&b` logical and
- `!a`

**Truth table**

| a | b | !a | a&&b | a||b |
|---|---|----|------|------|
| t | t | f  | t    | t    |
| t | f | f  | f    | t    |
| f | t | t  | f    | t    |
| f | f | t  | f    | f    |

**Practice**

What value does each boolean expression evaluate to?

```
int value1 = 5, value2 = 10;
boolean done = true;
1. value1 >= value2
   5>= 10  
   false
2. (value1+5) >= value2
   (5+5)>= 10
   true
3. value1 < (value2/2)
   5 < (10/2)
   false
4. value1 == value2
   5 == 10 false
5. !(value1 == value2)
   !false  
   true
6. -5<= value1 && value1<=5
   -5<= 5 && 5<=5
   true && true
   true
7. !(value1<value2)||done
   !(5<10) is false
   f || t 
   true
8. (value1>value2) | | !done
   (5>10) is false
```
!true is false
false || false = false
9. (value1>value2) && done
   (5>10) is false
   (value1>value2) && done false
10. done || !done
    done is true
    done || !done is true
11. ((value1>value2)|| done) && (!done||(value2>value1))
a && b
   a = (value1>value2)|| done
      = (5>10) || true
      = false || true
      = true
   b = (!done||(value2>value1))
      = !true || (10>5)
      = false || true
      = true
   a && b true

    public static double withdraw(double balance, double amount){
    if(balance < amount){
        System.out.println("Error: can not withdraw");
    } else {
        balance = balance - amount;
    } return balance;
    }

    public static double withdraw(double balance, double amount){
    if(balance < amount){
        System.out.println("Error: can not withdraw");
        return balance;
    } else {
        balance = balance - amount;
        return balance;
    }
    }

    Pratice:
1. Use conditions to write a method that returns random
   substrings of a word. Both the beginning and end locations
   should be random.
import java.util.Scanner;
import java.util.Random;
public class RandomString{
    public static void main(String[] args){
    }

    2. Write a method that determines the smallest of 3 integers