Comparing FP-Numbers
The Comparison Problem

• Given \( fp1 \) and \( fp2 \) of type float or double.

• Guideline 1:

  «Do not test two floating point numbers for equality, if at least one of them was rounded before.»
The Comparison Problem

- **Given** \( fp_1 \) and \( fp_2 \) of type `float` or `double`.

- **Guideline 1:**
  
  «**Do not** test two floating point numbers for equality, if at least one of them was rounded before.»

- **Thus** \( fp_1 == fp_2 \) **should be avoided.**
The Comparison Problem

• How can we compare instead?
The Comparison Problem

• How can we compare instead?

• First idea:
  Allow for small differences!

Given: tolerance value $c > 0$.

\[ \text{fp1 "equals" fp2 whenever } |\text{fp1} - \text{fp2}| < c \]

(Remark: $|...|$ means absolute value. In C++ it's not available using vertical bars.)
The Comparison Problem

Given: tolerance value \( c > 0 \).

\[ fp1 \text{ "equals" } fp2 \text{ whenever } |fp1 - fp2| < c \]

• Examples (\( c \) is 0.001):
  • \( fp1 = 10.0 \) and \( fp2 = 12.0 \)

(Remark: on this slide = is meant in the mathematical sense.)
The Comparison Problem

Given: tolerance value $c > 0$.

$fp1 \text{ "equals" } fp2$ whenever $|fp1 - fp2| < c$

• Examples ($c$ is $0.001$):
  • $fp1 = 10.0$ and $fp2 = 12.0$
    $|10.0 - 12.0| = 2.0$

(Remark: on this slide $=$ is meant in the mathematical sense.)
The Comparison Problem

Given: tolerance value $c > 0$.

$fp1$ "equals" $fp2$ whenever $|fp1 - fp2| < c$

• Examples ($c$ is 0.001):
  • $fp1 = 10.0$ and $fp2 = 12.0$
    $|10.0 - 12.0| = 2.0 > c$
    Thus: not "equal"

(Remark: on this slide = is meant in the mathematical sense.)
The Comparison Problem

Given: tolerance value $c > 0$.

$fp1 \ "equals" \ fp2$ whenever $|fp1 - fp2| < c$

- **Examples ($c$ is 0.001):**
  - $fp1 = 10.0$ and $fp2 = 12.0$
    - $|10.0 - 12.0| = 2.0 > c$
    - Thus: not "equal"

  - $fp1 = 10.0$ and $fp2 = 10.000013$

(Remark: on this slide $=$ is meant in the mathematical sense.)
The Comparison Problem

Given: tolerance value \( c > 0 \).

\[
\text{fp1 "equals" fp2 \ whenever \ } |\text{fp1} - \text{fp2}| < c
\]

- **Examples (c is 0.001):**
  - \( \text{fp1} = 10.0 \ 	ext{and} \ 	ext{fp2} = 12.0 \)
    \[
    |10.0 - 12.0| = 2.0 > c
    \]
    Thus: **not "equal"**

  - \( \text{fp1} = 10.0 \ 	ext{and} \ 	ext{fp2} = 10.000013 \)
    \[
    |10.0 - 10.000013| = 0.000013
    \]

(Remark: on this slide = is meant in the mathematical sense.)
The Comparison Problem

Given: tolerance value $c > 0$.

$fp1 \ "equals\" \ fp2$ whenever $|fp1 - fp2| < c$

• **Examples ($c$ is 0.001):**
  - $fp1 = 10.0 \ and \ fp2 = 12.0$
    - $|10.0 - 12.0| = 2.0 > c$
    - Thus: **not "equal"**

  - $fp1 = 10.0 \ and \ fp2 = 10.000013$
    - $|10.0 - 10.000013| = 0.000013 < c$
    - Thus: **"equal"**

(Remark: on this slide $=$ is meant in the mathematical sense.)
Exercise

Write the following function:

```java
// POST: returns true if and only if
//       |x - y| < tol
bool equals (double x, double y, double tol) {
    ...
}
```
Exercise

For example:

```c
// POST: returns true if and only if
//       |x - y| < tol
bool equals (double x, double y, double tol) {
    double diff = x - y;
    if (diff < 0)
        diff *= -1; // absolute value
    return diff < tol;
}
```
Remark

• Comparing absolute differences with a tolerance value is a great first idea!

• (But: for example problems when the numbers are large.)