

Floating Point Guidelines

Guidelines

Guideline 1:

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

Guideline 1 – Example

Guideline 1:

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



This is *false*

Example:

```
double a = 1.1;
if (100 * a == 110)
    std::cout << "no output\n";
```

Guideline 1 – Example

Guideline 1:

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

This is false

Example:

```
double a = 1.1;  
if (100 * a == 110)  
    std::cout << "no output\n";
```

Problem:

1.1 not
representable

Guideline 1 – Example

Guideline 1:

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

This is false

Example:

```
double a = 1.1;  
if (100 * a == 110)  
    std::cout << "no output\n";
```

Problem:

1.1 not
representable

1.1 = $\overbrace{1.0001100110011001100110011}^{24\text{bit}}$
(rounding) $\rightarrow 1.10000002384\dots = 1.00011001100110011001101$

Guidelines

Guideline 1:

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

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes!**»

Guideline 2 – Example

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes!**»

Example:

```
float a = 67108864.0f + 1.0f;  
  
if (a > 67108864.0f)  
    std::cout << "This is not output ... \n";
```

Guideline 2 – Example

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes!**»

Example:

```
float a = 67108864.0f + 1.0f;  
  
if (a > 67108864.0f)  
    std::cout << "This is not output ... \n";
```

Problem:

Significand too
short

Guideline 2 – Example

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes!**»

Example:

```
float a = 67108864.0f + 1.0f;  
  
if (a > 67108864.0f)  
    std::cout << "This is not output ... \n";
```

Problem:

Significand too
short

$$\begin{array}{r} \\ \\ \hline 67108864 = \overbrace{1.000000000000000000000000}^{24\text{bit}} \cdot 2^{26} \\ +1 = 0.00000000000000000000000000000001 \cdot 2^{26} \\ \hline 67108865 = 1.000000000000000000000001 \cdot 2^{26} \end{array}$$

Guideline 2 – Example

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes!**»

Example:

```
float a = 67108864.0f + 1.0f;  
  
if (a > 67108864.0f)  
    std::cout << "This is not output ... \n";
```

Problem:

Significand too short

$$\begin{array}{r} \\ \\ \\ \text{(rounding)} \rightarrow 67108864 \end{array} = \begin{array}{r} \\ \\ \\ 1.000000000000000000000000 \end{array} \cdot 2^{26}$$

24bit

Guidelines

Guideline 1:

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

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes!**»

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes!**»

Guideline 3 – Example

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes!**»

Example:

- Consider sequence $x_{n+1} = 6x_n - 1$

Guideline 3 – Example

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes!**»

Example:

- Consider sequence $x_{n+1} = 6x_n - 1$
- Computing some sequences for given x_0 :

Guideline 3 – Example

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes!**»

Example:

- Consider sequence $x_{n+1} = 6x_n - 1$
- Computing some sequences for given x_0 :
 - e.g. $x_0 = 1 \quad \rightarrow \quad x_1 = 5, \quad x_2 = 29, \quad x_3 = 173, \quad \dots$

Guideline 3 – Example

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes!**»

Example:

- Consider sequence $x_{n+1} = 6x_n - 1$
- Computing some sequences for given x_0 :
 - e.g. $x_0 = 1 \quad \rightarrow \quad x_1 = 5, \quad x_2 = 29, \quad x_3 = 173, \quad \dots$
 - e.g. $x_0 = 0.2 \quad \rightarrow \quad x_1 = 0.2, \quad x_2 = 0.2, \quad x_3 = 0.2, \quad \dots$

Guideline 3 – Example

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes!**»

Example:

- Consider sequence $x_{n+1} = 6x_n - 1$
- Computing some sequences for given x_0 :
 - e.g. $x_0 = 1 \quad \rightarrow \quad x_1 = 5, \quad x_2 = 29, \quad x_3 = 173, \quad \dots$
 - e.g. $x_0 = 0.2 \quad \rightarrow \quad x_1 = 0.2, \quad x_2 = 0.2, \quad x_3 = 0.2, \quad \dots$

C++ claims

$x_{14} \approx 622.982$

Guideline 3 – Example

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes!**»

Example:

- What went wrong?

Guideline 3 – Example

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes!**»

Example:

- What went wrong?
 - `float` represents 0.2 as 0.20000000298...
 - Thus: $6 \cdot x_0 - 1 \neq 1.2 - 1$

Guideline 3 – Example

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes!**»

Example:

- What went wrong?
 - `float` represents 0.2 as 0.20000000298...
 - Thus: $6 \cdot x_0 - 1 \neq 1.2 - 1$ but rather:
 - $x_1 = 0.20000004768 \dots$
 - $x_2 = 0.20000028610 \dots$
 - $x_3 = 0.20000171661 \dots$
 - \vdots

Guideline 3 – Example

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes!**»

Example:

- What went wrong?
 - `float` represents 0.2 as 0.20000000298...
 - Thus: $6 \cdot x_0 - 1 \neq 1.2 - 1$ but rather:
 - $x_1 = 0.20000004768 \dots$
 - $x_2 = 0.20000028610 \dots$
 - $x_3 = 0.20000171661 \dots$
 - \vdots

Note how error increases!