Programming Mathematics

# Lab 02

Save all work to a Lab 02 folder on your shared folder in your dropbox.

# Number Systems Continued...

## Floating Point Numbers.

Find a recurring fraction f say, e.g. ... (Do not pick !!) Estimate the round off error for each of the following using floats and doubles by any means.

Outline how you do it here

**Method:** I have a lab2.java file provided to calculate the float and double using loops and Strings. I provided necessary documentation (comments) within my code to show how I achieved the outlined results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Float using (loop) | Float using (String comparison) | Double using (String) | Double using (Loop) |
| F = 3/7 | -2.5285994E-5 | 4.2574746285714285714285714285714e-9 | -2.1428571428571428571428571428571e-17 | 1.1693503308509077E-14 |
| 10f | -1.5178572E-4 | 4.2574746285714285714285714285714e-9 | 2.8571428571428571428571428571429e-16 | -3.305363601871899E-13 |
| 100f | -3.9285715E-4 | 5.4495675285714285714285714285714e-7 | -3.1428571428571428571428571428571e-15 | -2.7440754430634637E-12 |
| 1000f | -0.004857143 | 385.71429879324774285714285714286 | -1.1428571428571428571428571428571e-14 | -3.552330391747611E-11 |
| 100000 + f | 100007.04 | 0.00111607142857142857142857142857 | -1.4285714285714285714285714285714e-12 | 99999.99999999923 |
|  | 2.9300363E-5 | 0.04285714285714285714285714285714 | 2.8571428571428571428571428571429e-18 | -4.2798222109143226E-14 |
|  | 2.6501117E-5 | 0.00428571428571428571428571428571 | 2.8571428571428571428571428571429e-19 | 2.1307511945321624E-14 |
|  | 3.8830636E-5 | 4.2857142857142857142857142857143e-4 | -2.1428571428571428571428571428571e-20 | -3.4733992444151745E-14 |
|  | -2.8228571E-5 | 4.2857142857142857142857142857143e-5 | -1.1428571428571428571428571428571e-21 | 5.0299130245028726E-14 |
|  | 31414.617 |  | -2.8571428571428571428571428571429e-12 | 99999.99996102002 |

**Comment:**

1. **String is calculated by dividing f = n/d in the calculator.**
2. **Double is more precise.**
3. **Bigger the number, bigger the error.**

**(10 + ) \* 3 = 30 + 1 =31**