Volume 2, Issue 9, September 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Case Study of Floating Point Value Problem

Manoj Mukherjee
Dept. of BCA (Hons)
Burdwan Institute of Management & Computer Science
Dewandighi, Katwa Road, Burdwan - 713102, West Bengal, India

Abstract- Programming language of computer science plays a major role to solve a specific real life problem. But in the programming language we faced a specific problem in subtraction and addition of a few specific floating numbers. The result of those specific numbers does not give a correct answer. This problem is faced in some major programming languages. The challenge for us is to overcome this problem. This paper highlights the problem and provides some possible approach to this problem.

Keywords— Floating Value, Floating point.

I. INTRODUCTION

We know that every variable has a data type. Data types specific the size and types of values that can be stored. The varieties of data types available allow the programmer to select the type appropriate to needs of the application. In many programming language common data type are integer, floating-point and character etc.

II. INTEGER TYPES

Integer types can hold whole numbers such as 145,-45, and 4512. In java programming language supports four types of integers [2]. They are byte, short, int and long.

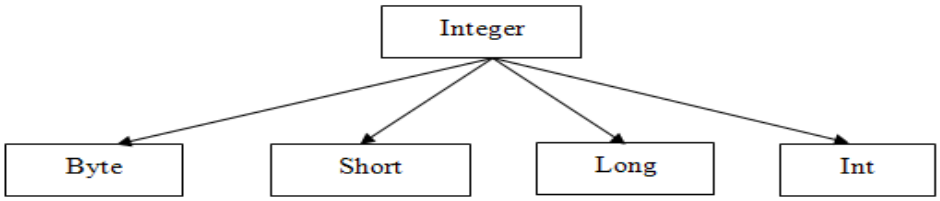


Fig: Integer data types

Table: Size and Range of Integer Types in Java

Type	Size	Minimum value	Maximum value
byte	One byte	-128	127
short	Two byte	-32,768	32,767
int	Four byte	-2,147,483,648	-2,147, 483,647
long	Eight bytes	-9,223,372, 036,854,775,808	-9,223,372, 036,854,775,807

III. FLOATING POINT TYPES

Integer types can hold only whole numbers and therefore we use another type known as floating point type to hold numbers containing fractional parts such as 27.59 and -1.375 [3].

The float type values are single-precision numbers while the double type represents double-precision numbers [1].

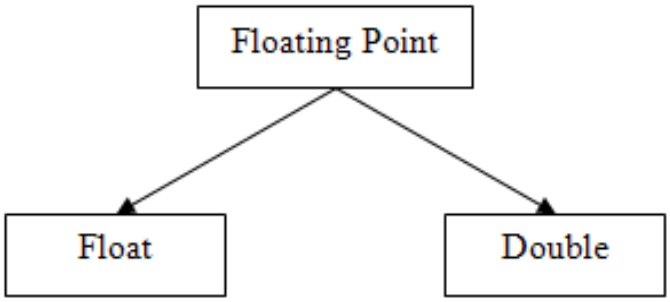


Fig: Floating point data types

Type	Size	Minimum value	Maximum value
float	4 bytes	3.4e-038	1.7e+0.38
double	8 bytes	3.4e-038	1.7e+308

Floating point data types support a special value known as Not-a-Number (NaN).

IV. CHARACTER TYPE

In order to store character constant in memory. A character data type char. It can hold only 1 bytes.

V. PROBLEM NITIALIZATION

In 'C programming language'

Program Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
float a,b,sum,sub;
clrscr();
```



```
printf("Enter the floating point value for addition and substraction a and b\n");
scanf("%f %f",&a,&b);
sum=a+b;
printf("\naddition result a and b is %f",sum);
sub=a-b;
printf("\nSubstraction result a and b is %f",sub);
getch();
}
```

VI. OUTPUT IN DIFFERENT FLOATING VALUES

1.

```
Enter the floating point value for addition and substraction a and b
10.05
20.05

addition result a and b is 30.099998
Substraction result a and b is -9.999999_
```

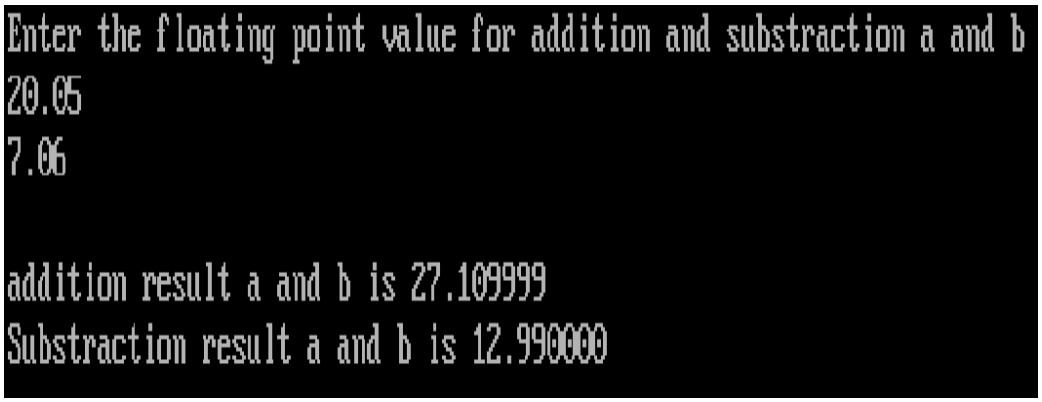
2.

```
Enter the floating point value for addition and substraction a and b
50.05
60.05

addition result a and b is 110.099998
Substraction result a and b is -10.000000_
```

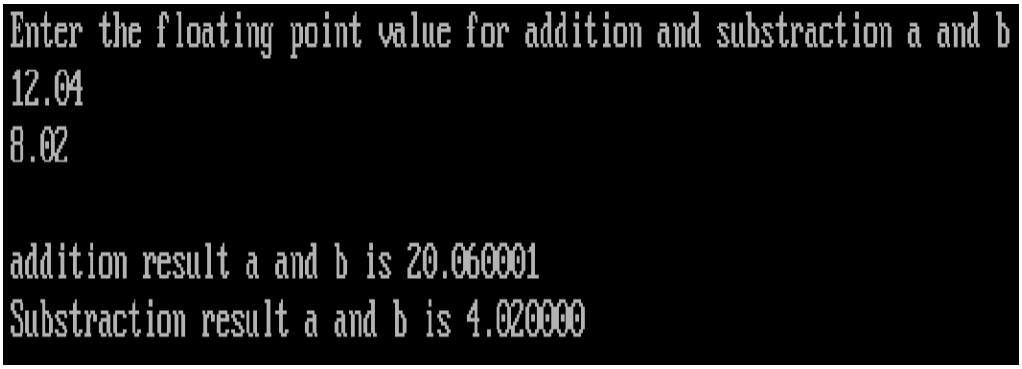
3.

```
Enter the floating point value for addition and substraction a and b
80.09
60.10

addition result a and b is 140.190002
Substraction result a and b is 19.989998
```


4.



5.



In 'JAVA programming language'

```
import java.util.Scanner;
class ADD
{
	public static void main(String arg[])
	{
	float x;
	float y;
	Scanner in=new Scanner(System.in);
	System.out.println("Enter an integer for x");
		x=in.nextFloat();
	System.out.println("Enter an integer for y");
		y=in.nextFloat();
	System.out.println("The ADD:: "+(x+y));
	System.out.println("The SUB:: "+(x+y));
	}
}
```

OUTPUT:

1.

```
----------Configuration: <Default>----------
Enter an integer for x
```


10.05
Enter an integer for y
20.05
The ADD:: 30.099998
The SUB:: -9.999999

Process completed.

2.

----------Configuration: <Default>----------
Enter an integer for x
1.03
Enter an integer for y
29.02
The ADD:: 30.050001
The SUB:: -27.99

Process completed.

3.

----------Configuration: <Default>----------
Enter an integer for x
13.6
Enter an integer for y
17.4
The ADD:: 31.0
The SUB:: -3.7999992

Process completed.

4----------Configuration: <Default>----------

Enter an integer for x
15.03
Enter an integer for y
14.02
The ADD:: 29.05
The SUB:: 1.0099993

Process completed.

5.

----------Configuration: <Default>----------
Enter an integer for x
11.03
Enter an integer for y
19.03
The ADD:: 30.060001
The SUB:: -8.000001

Process completed.

VII. PROPOSE WAY TO SOLUTION

Generally, a float has about 7 digits precision. So 30.099998 is correct to 7 digits.
The reason it does not give the exact answer a noob would expect is because the number is actually processed in binary floating point and just like (1/3)x3 will give you 0.9999999 if you do the math and only allow 7 digits, so 0.05 is represented as a repeating fraction in binary and so is only close to the actual value of 0.05.
Do you know how to convert 0.05 to a binary fraction? Well 0.5 = 2^-1 and 0.25 = 2^-2 and 0.125 = 2^-3, etc. So just like you build up integers in binary, binary fractions are built up (although "built down" might be a better way of thinking about it) by finding the sum of negative powers of 2 that add up to it. First term is 0.03125 and 0.5 - 0.03125 = 0.016825 which is less than the next power of 2 so the first 6 digits are 0.000101 0.016825- 0.015625 = .001200 which is more than the next power so
0.0001011,etc.

Decimal					Binary
0.5.	.	.	.	.	0.1
0.25	.	.	.	.	0.01
0.125	.	.	.		0.001
so	0.875		=		0.111

(and guess what a binary 0.11111111 (1 repeats) equals?)

VIII. CONCUSSION

The problem highlighted is discussed in details with examples. Some possible measures to overcome this problem is also proposed in above writing. Hope this may help in further research on this topic and finally solve it.

ACKNOWLEDGMENT

Thankful to our institution, dept. of BCA (Hons) Burdwan Institute of Management & Computer Science,Dewandighi, Katwa Road, Burdwan - 713102, West Bengal, India and project guide , Mr. Debabrata Samanta, Asst. Professor, Dept of Computer Application Burdwan Institute of Management and Computer Science, Burdwan, West Bengal, India.

REFERENCES

[1] http://digitaljournal.com/article/312789

[2] http://www.cygnus-software.com/papers/comparingfloats/comparingfloats.htm

[3]http://www.cprogramming.com/tutorial/floating_point/understanding_floating_point_representation.html

