Floating Point Math

Submitted by Andy Lindsay on Thu, 03/21/2013 - 16:37

original source: http://learn.parallax.com/propeller-c-start-simple/floating-point-math

Lesson edited to work with **Dev-C++** IDE by Jeff La Favre 10/22/15

(Updated 2013-08-08 for SimpleIDE 0.9.4 and its Learn folder's Simple Libraries and Examples) [SimpleIDE is the IDE for use with the robot. This lesson is edited so that we can use ***Dev-C++*** *as the IDE, which does not require a robot – J. La Favre]*

So far, these lessons have processed integers values, which encompass negative and positive counting values. C language also handles *floating point* values, which allow you to process numbers with a decimal point and one or more digits to the right, much as a calculator does. Since the number of digits to the left or right of the decimal point is flexible, the decimal point's position can "float" from one position to another as needed.

- Use a calculator and c = 2×π×r to calculate the circumference of a circle with a radius of 1.0.
- Start **Dev-C++**.
- Open the **File** menu and select **New**. Then select **Source File**.
- Click the mouse in the text window of **Dev-C++** and use the keyboard to enter the following text: **#include <stdio.h>**
- Open the **File** menu and select **Save**, which opens a **Save As** dialog box.
- In the dialog box, open the drop-down labeled **Save as type** and select **c source files(*.c)**. In the **file name** slot enter this name for the file: **floating point calculations** At the top of the dialog box there is a **Save in** slot, which determines where the file will be saved. Make sure you know the location where you are saving your file so that you can find it later. Now click the **Save** button to save your program file.
- Copy the text in the box on the next page and paste it into the text window of **Dev-C++** under the first line of text you have already entered. Alternatively, you can enter the text using the keyboard (enter exactly as given, without comments).

- Click the **Save** button to save the code you just pasted or entered with keyboard.
- Now examine the program code, and try to predict what will be displayed when the program runs.
- Run the program by opening the **Execute** menu and selecting **Compile and Run**. If there are no errors in the program, a new program window will open. Compare the actual output to your predicted output.

```
#define   PI   3.14 //create a constant named PI and give it a value of 3.14, notice you don't use an = operator in this case

int main()

{  //begin of code block for main()

  float r = 1.0;      //create a variable named r of the floating point data type and assign a value of 1.0 to it

  float c = 2.0 * PI * r;     //create a float type variable named c and store the answer of the calculation in it

  printf("circumference = %f \n", c);  // in this case the following will be displayed on screen: circumference = 6.280000

}  // end of code block for main()
```

How it Works

[This program knows that **`PI`** ≈ 3.1415926... because it is defined in the simpletools library. – Ed. Note: this is true for the IDE for the robot, but not for Dev-C++, therefore we need to define PI as a constant, which is done with the line **#define PI 3.14**]

Inside the **`main`** function, a floating point variable named **`r`** is initialized to 1.0 with **`float r = 1.0`**.

After that, the circumference is calculated with
`c = 2.0 * PI * r`.

Then, **`printf("circumference = %f \n", c)`** displays the floating point value stored in **`c`**. Notice the new format placeholder: **`%f`** for a floating point value.

Did You Know?

You can declare different variable types that can store different sizes and types of numbers.

signed char	-127 to 127
char	0 to 255
int	-2,147,483,647 to 2,147,483,647
unsigned int	0 to 4,294,967,295
long	same as **int**
unsigned long	same as **unsigned int**
float	approx: $3.4X10^{-38}$ to $3.4X10^{38}$ with 6 digits of precision

Try This

Let’s try calculating the area of a circle with a = $\pi \times r^2$, which is PI * r * r.

- Use **Save As** in the **File** menu to save your program with the name **More Floating Point** (make sure to save it as a **c source files(*.c)** type.)
- Modify the **`main`** function by adding the two statements under the first printf.
- Compare the program result against your calculated result.
- Try changing the radius to 3 and repeat.

```
#define   PI   3.14 //create a constant named PI and give it a value of 3.14, notice you don’t use an = operator in this case

int main()

{

  float r = 1.0;

  float c = 2.0 * PI * r;

  printf("circumference = %f \n", c);

 float a = PI * r * r; //create a float type variable named a and store the answer of the calculation in it

 printf("area = %f \n", a);

}
```

Your Turn

- Expand More Floating Point.c to calculate the volume of a sphere using $v = 4/3 \times \pi \times r^3$.

Keep in mind that you have to use 4.0/3.0 to get the floating point version of 4/3. You can also use pow(r, 3.0) to raise r to the third power.