# Mathematical Literacy – Part 3

# A Challenge from my Blog

Sunday, December 01, 2013

## A Rs. 1,00,00,000 Offer from me , If you are able to find a triplets ( 3 #'s satisfying a mathematical property )

When I learned to write computer programs,finding Pythagorean triplets was one of the first programs written by me. One can easily find lot of triplets which satisfy Pythagorean triangle equation , $x^2 + y^2 = z^2$.

Some examples are , $4^2 + 3^2 = 5^2$

$12^2 + 5^2 = 13^2$

The challenge is to find out triplets for any number n , provided n > 2 which satisfies the equation

$x^n + y^n = z^n$.

If you are able to find a integer solution, you will get this reward !

# Pythagorean Triplets in Python

```
from math import sqrt
n = int(10)+1
print("-----------------------------------\n")
for a in range(1,n):
   for b in range(a,n):
      c_square = a**2 + b**2
      c = int(sqrt(c_square))
      if ((c_square - c**2) == 0):
         print(a, b, c,"\n")
```

## Words which gave Mathematicians Sleepless nights for Centurites

The Celebrated French Mathematician wrote the following in one of his book.

**"It is impossible to separate a cube into two cubes, or a fourth power into two fourth powers, or in general, any power higher than the second into two like powers. I have discovered a truly marvelous proof of this, which this margin is too narrow to contain."**

What essentially, he told was

$$x^3 <> y^3 + y^3 \text{ or } x^4 <> y^4 + z^4,$$
$$\text{in general, } x^n + y^n <> z^n \text{ for } n > 2$$

We are familiar with Pythagores Therom, whcih states that,

"In a right triangle, hypotenuse (z ) squared is equivalent to base (x) squared added to opposite side(y)"
Algebraically, It can be written as,

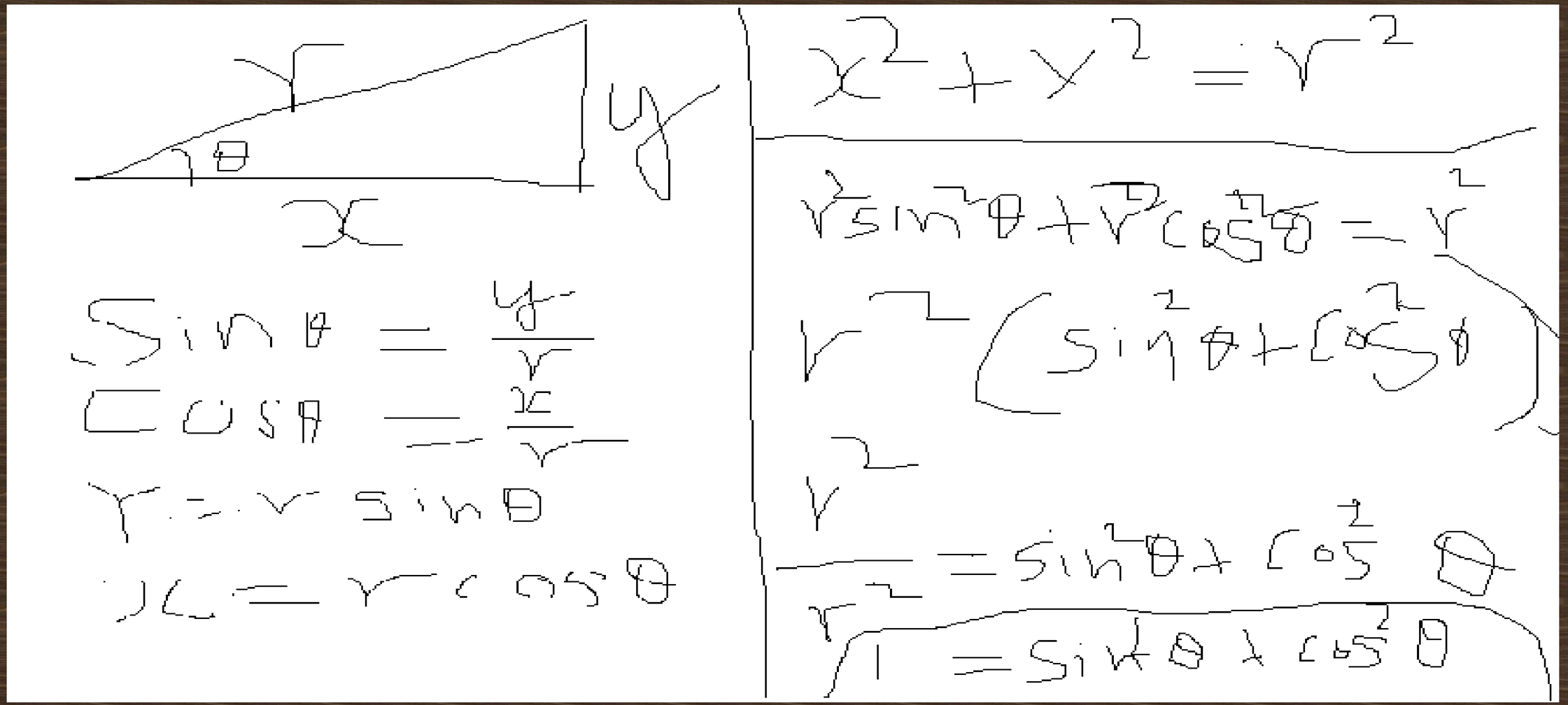
$$z^2 = x^2 + y^2.$$

Diophantine equations are generalization of Pythagoras theorem where factors (x,y,z) are integral numbers. What Fermat essentially told us was , **For a Diophantine equation with powers greater than 2, there is no solutions.**

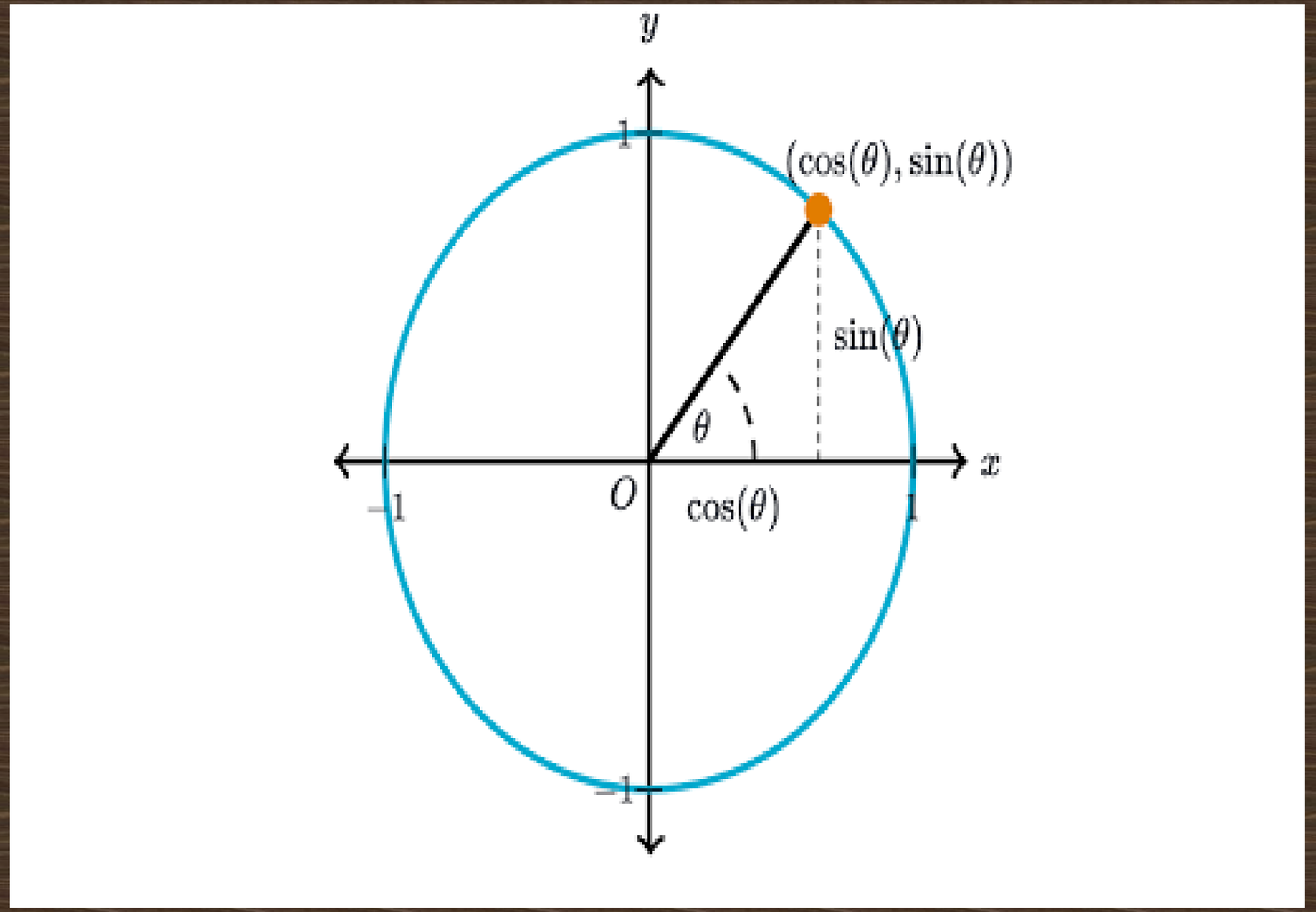
# Pythagorean Triplets and Testing FLT

```
#include <stdio.h>
int main() {
float x , y , z ;
for( x = 1.0 ; x <= 100; x=x+1.0 )
 for( y = 1.0 ; y <= 100 ; y=y+1.)
  for( z = 1.0 ; z <=100; z=z+1. ){
   if ( x*x + y*y == z*z )
      printf(" triplets %d\t\t%d\t\t%d\n",
                                  (long)x,(long)y,(long)z);
   if ( x*x*x + y*y*y == z*z*z )
         printf("cube =%d\t\t%d\t\t%d\n",
                                  (long)x,(long)y,(long)z);
  }
}
```

# A wonderful proof



# Trignometry



# Why do Software Libraries ask you to give angle in radians ?

Trigonometric functions are evaluated using a series ( a variant of Taylor series you studied in Calculus ) where the arguments are supposed to be in radians !. The following series can help you to evaluate Sine and Cosine.

$$\begin{aligned}
\sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \\
&= \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}, \\
\cos x &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \\
&= \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}.
\end{aligned}$$

One can define a Macro or function to convert Degrees to Radians

```
double DTOR( double angle ) {
    return angle*3.14159/180.0
}
```

# How to Compute Sine Theta ?

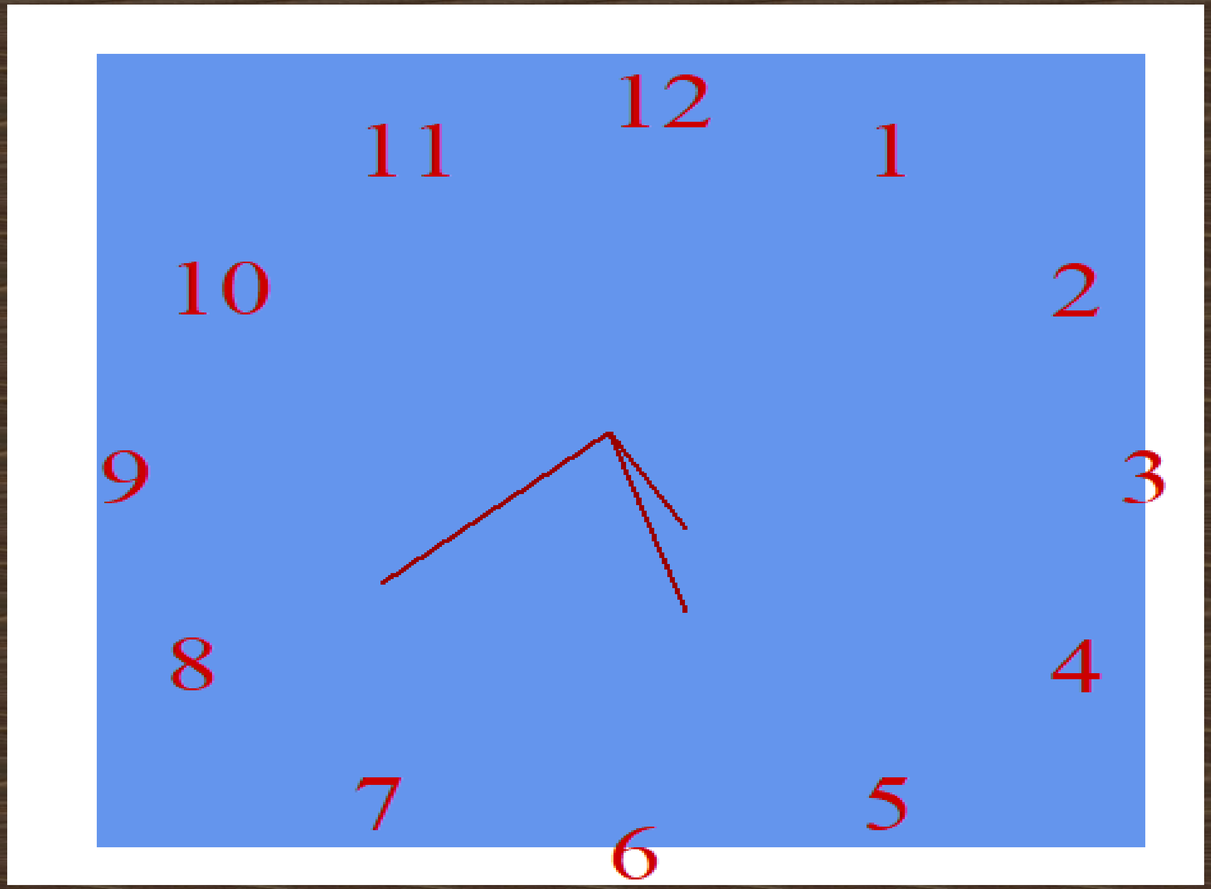
```
#include <stdio.h>
#include <math.h>

double m_abs( double arg ) {
        if ( arg < 0.0 ) { arg = -1*arg;}
        return arg;
}

double m_dtor(double x ) { return ( x * 3.14159/180.0 ); }

double m_sin(double arg ) {
        double term = arg;
        double i = 1;
        double x2 = arg*arg;
        double tsin = arg;
        while ( m_abs(term) > 0.0000001 ){
         i = i + 2;
              term = -term*x2/(i*(i-1));
         tsin = tsin + term;
        }
        return tsin;
}
```

```
double m_cos(double arg ) { return m_sin( (2*3.14159/4.0) - arg ); }

int main(int argc, char **argv) {
  double dval = (argc == 1 )? 45.0 : atof(argv[1]);
  printf("\n\n\n\nValue is %e\n\n" ,dval*3.14159/180.0);
  printf("my sine = %f\t\t\tmath.h sin = %f\n",
                    m_sin(m_dtor(dval)),sin(m_dtor(dval)));
  printf("my cosine = %f\t\t\tmath.h cosine = %f\n",
                    m_cos(m_dtor(dval)),cos(m_dtor(dval)));

}
```

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}$$

# How To Render a Clock?



# How To Convert Co-ordinate System?

```
const double PI = 3.14159;

void Transform(HWND m_hWnd, int *px , int *py){
	RECT _rect;
	::GetClientRect(m_hWnd,&_rect);
	int width = (_rect.right-_rect.left)/2;
	int height = (_rect.bottom-_rect.top)/2;
	*px = *px + width;
	*py = height - *py;
}
```

# The Render Routine

```
void CallPaintRoutine( HWND hwnd , HDC paintdc ){
    int x1 = -200,y1=-220,x2 = 210,y2=200;
    Transform(hwnd,&x1,&y1); Transform(hwnd,&x2,&y2);
    RECT rect = { x1,y1,x2,y2};
    HBRUSH hbr = CreateSolidBrush(RGB(100,149,237));
    FillRect(paintdc, &rect,hbr);
    HPEN hpen = CreatePen(PS_SOLID,2,RGB(153,0,0));
    SelectObject(paintdc,hpen);
    SYSTEMTIME systime;GetLocalTime(&systime);

    int fhour,fmin,fsec ;
    fhour = systime.wHour%12; fmin = systime.wMinute;
    fsec = systime.wMinute;

    fhour += fmin/60; fhour = fhour * 360 / 12;
    fmin = fmin * 360 / 60; fsec = fsec * 360 / 60;

    int xs =  120 * cos((-fsec * PI / 180.0)+ PI/2.0);
    int ys =  120 * sin((-fsec * PI / 180.0)+ PI/2.0);
    int xm =  100 * cos((-fmin * PI / 180.0)+ PI/2.0);
    int ym =  100 * sin((-fmin * PI / 180.0)+ PI/2.0);
    int xh =  60 * cos((-fhour * PI / 180.0)+ PI/2.0);
    int yh =  60 * sin((-fhour * PI / 180.0)+ PI/2.0);
```

```
    int x = 0,y=0;
    Transform(hwnd,&xh,&yh);Transform(hwnd,&x,&y);
    MoveToEx(paintdc,x,y,NULL);LineTo(paintdc,xh,yh);
    Transform(hwnd,&xm,&ym); MoveToEx(paintdc,x,y,NULL);
    LineTo(paintdc,xm,ym);Transform(hwnd,&xs,&ys);
    MoveToEx(paintdc,x,y,NULL);LineTo(paintdc,xs,ys);
    char *str[] = {"3","2","1","12","11","10","9","8","7","6","5","4"};
    HFONT hFont = CreateFont(48,0,0,0,FW_DONTCARE,FALSE,TRUE,FALSE,
            DEFAULT_CHARSET,OUT_OUTLINE_PRECIS,
            CLIP_DEFAULT_PRECIS,CLEARTYPE_QUALITY,
            VARIABLE_PITCH,TEXT("Times New Roman"));
    SelectObject(paintdc,hFont);
    SetTextColor(paintdc,RGB(204,0,0));
    SetBkMode(paintdc,TRANSPARENT);
    for(int i = 0;i < 12;i++){
       int x = 200 * cos(i * PI / 6),y= 200 * sin(i * PI / 6);
       Transform(hwnd,&x,&y);
       TextOut(paintdc,x,y,str[i],strlen(str[i]));
    }

}
```

# Q&A

- If any!
- Source Code of the Clock Program Available @ https://github.com/praseedpai/ElementaryMathForProgrammingSeries/blob/master/AlgebraNArith/AnalogClock/PSClock.cpp
- Source Code of the SineCos from Scratch available @ https://github.com/praseedpai/ElementaryMathForProgrammingSeries/blob/master/AlgebraNArith/SinCosScratch/SinCos.cpp
- Source Code of the Pythagorean and FLT available @ https://github.com/praseedpai/ElementaryMathForProgrammingSeries/blob/master/AlgebraNArith/Fermat/Pyth.cpp