

COBOL

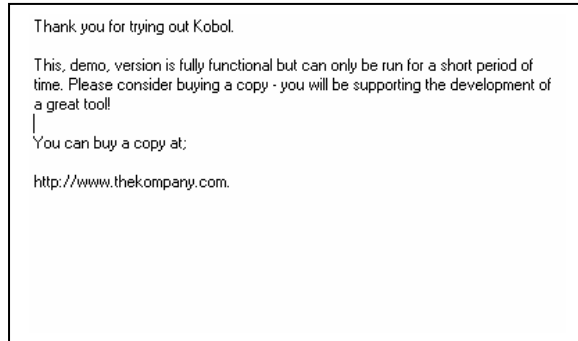
The only practical PC version I could locate, that worked (several ones I found did not support Windows XP) is called:- KOBOL

Available from:- www.thekompany.com/products/kobol/

This package is available for under \$100 usd, the trial version seems fully functioned but closes down after a few minutes. This is enough time to copy and paste a program into the IDE and then run it.

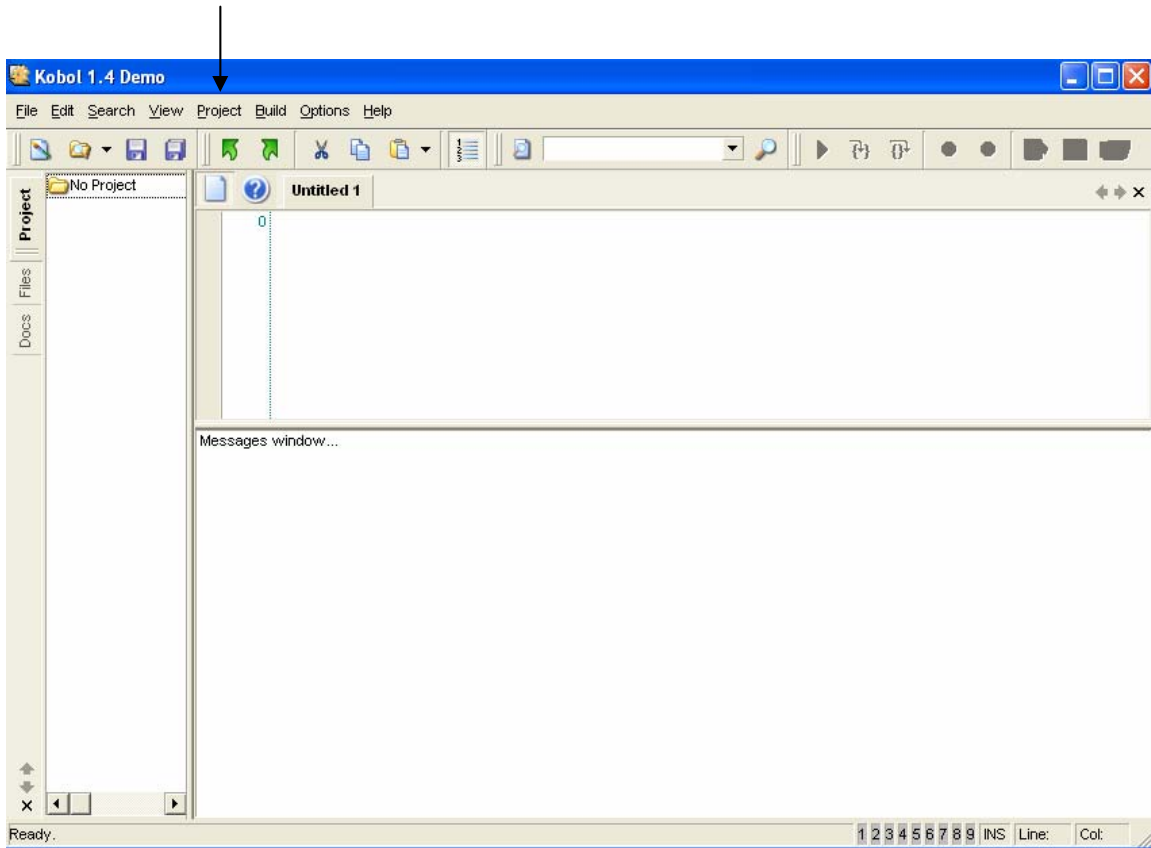
The system is not intuitive. Even the coding errors are not intuitive and it takes a while to figure out why the compiler objects to a line of code.

On first starting there is a greeting.



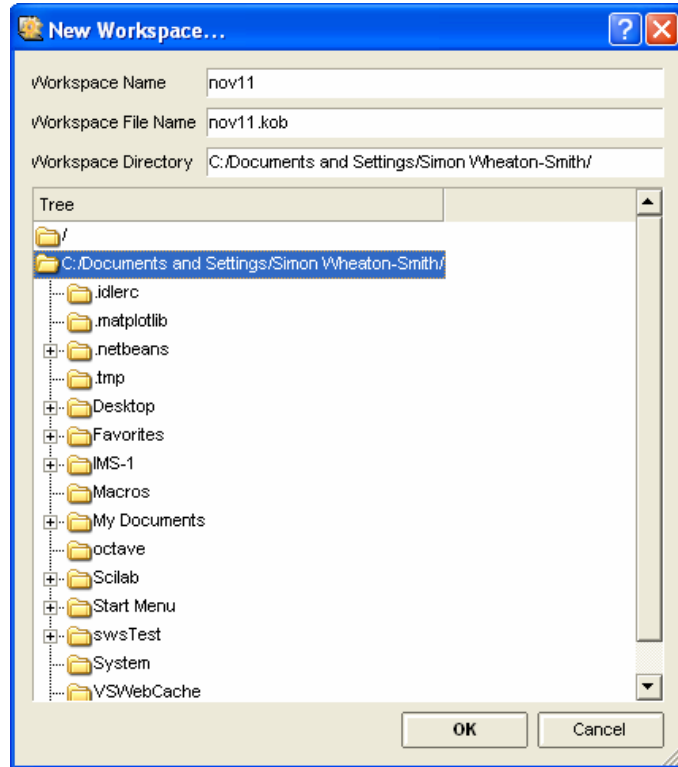
Then the main IDE arrives.

Nothing works without a PROJECT which is the 5th button over on the right at the top,

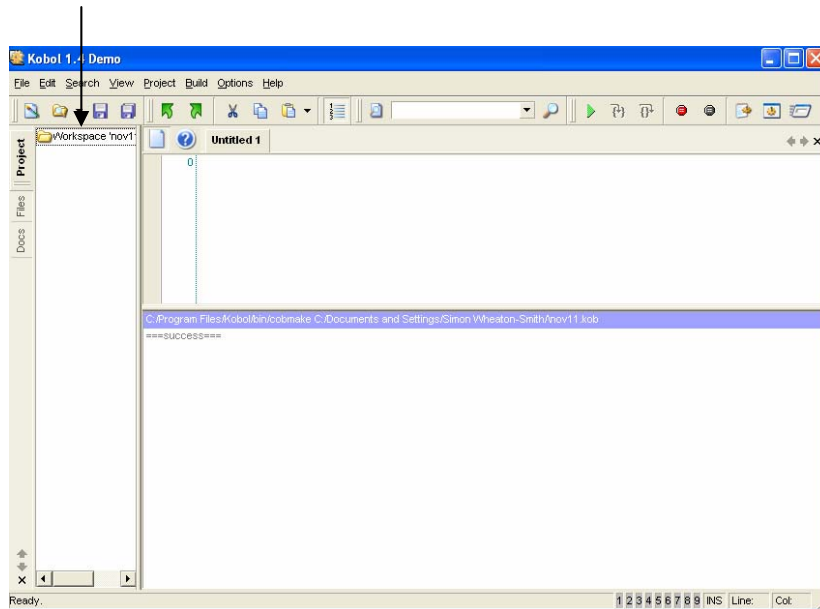


COBOL sundial programming notes purely for legacy reasons

So click on: PROJECT and then on NEW WORKSPACE and enter a workspace name. It generates a .KOB file name, however, you have to tell it where to store the project files. So, click on a choice in the lower part of the panel, and it will fill in the third field at the top of this project menu.



At this point, the new project appears in the top right panel. So far, this is only the project stuff, no programs.



COBOL sundial programming notes purely for legacy reasons

To enter a program, bring up the source code in something like WordPAD or NotePAD, and SELECT ALL (ctrl-a) and then copy it to the clip board (ctrl-c).

Then back in KOBOL, do the following:

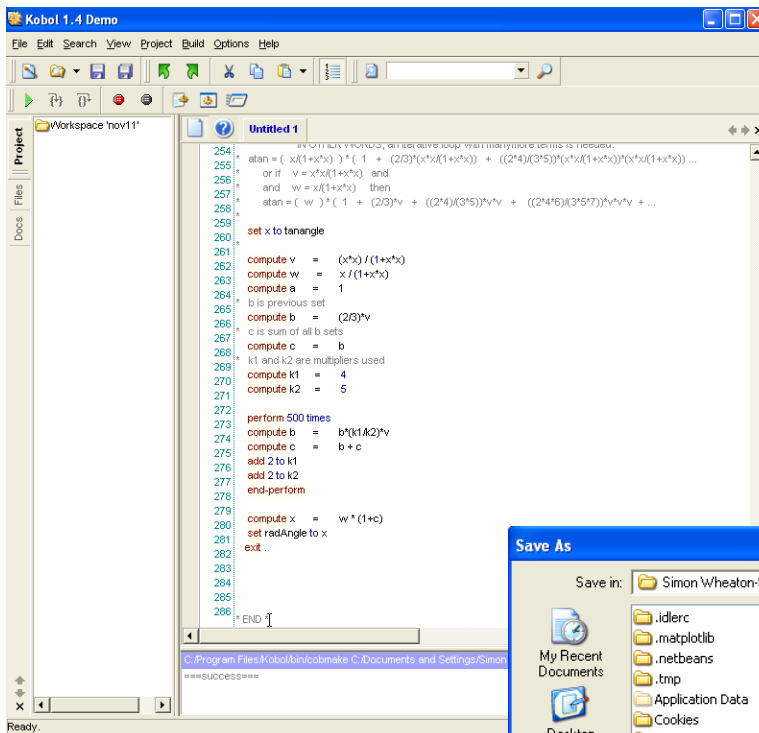
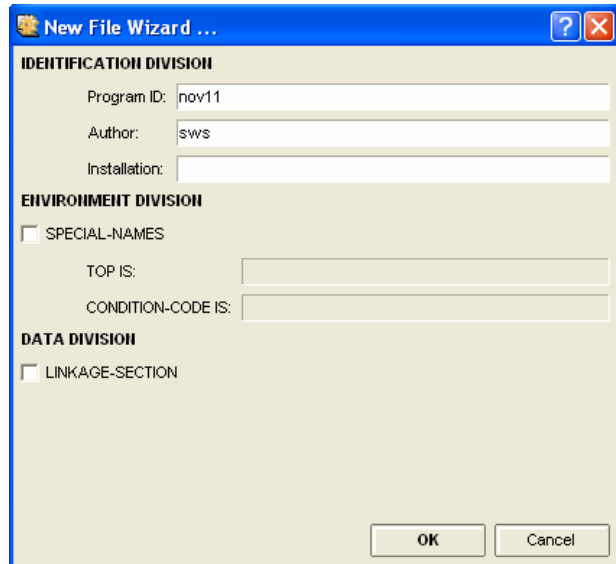
FILE
NEW

and fill in the name and the author. This will then generate a model program.

However, select the entire model program, all of it, and delete it.

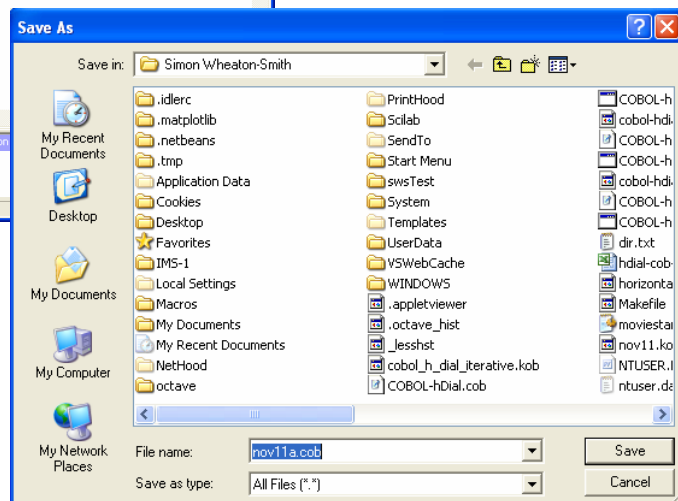
The system sometimes crashes here.

Then paste the copied program into this space (ctrl-v). And then save it.



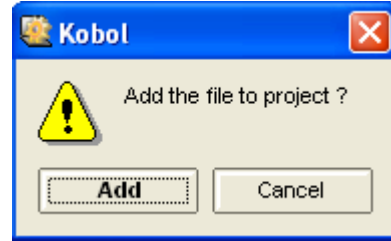
The program to the left is saved using FILE, SAVE AS, and it asks for a name. Give it a name and make sure you add the .COB suffix.

Remember, .KOB is a project file and .COB is a program.



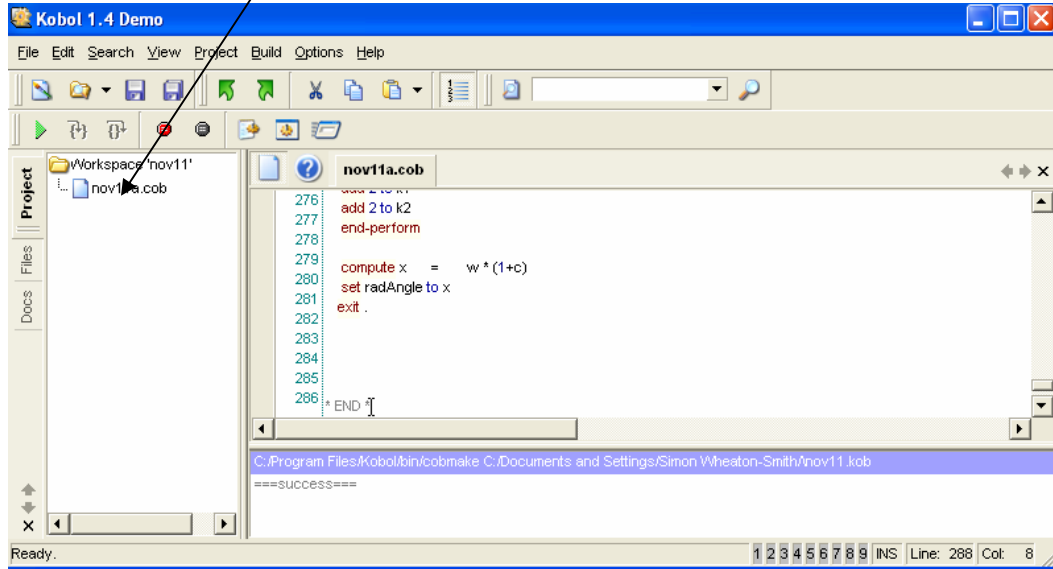
COBOL sundial programming notes purely for legacy reasons

When the save is completed, it will ask you if you want to add it to the project.



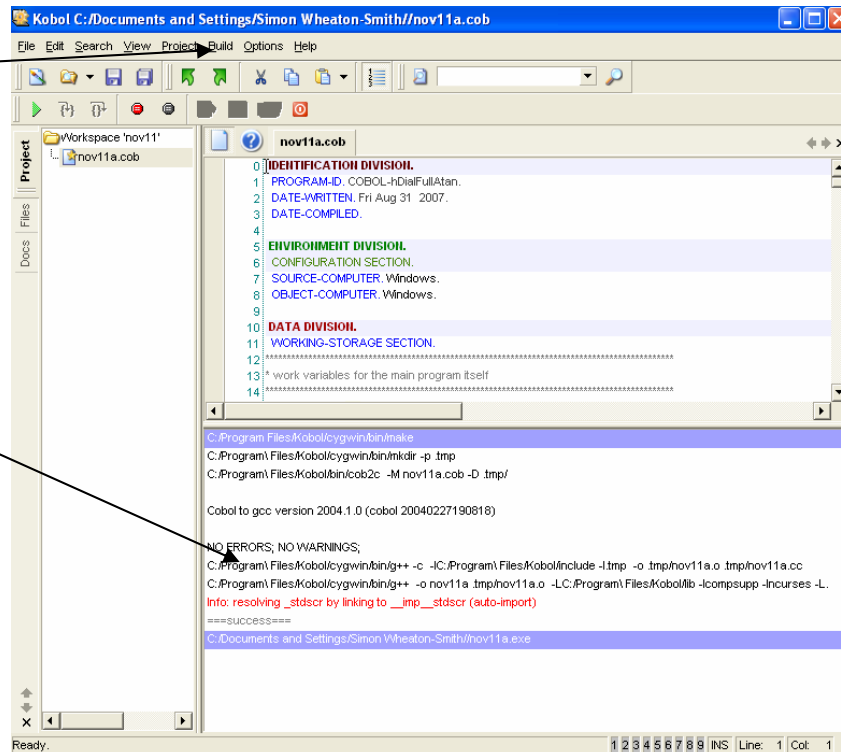
And you click on ADD.

The program then appears in the project panel.



Then you click on BUILD

The BUILD then RUN will compile the program and then run it. Compile and link messages will appear in the lower right box.



The program then runs in a DOS window that Windows brings up.

```

[COBOL-hdial.cob]
function ATAN is valid for angles<80
Enter a latitude
32.75
Enter a longitude
108.2
Enter reference longitude
105

Latitude, longitude, reference
LAT: 0032.75000  LNG: 0108.20000  REF: 0105.00000

correction in minutes: 0012.80000

Hr: 06.0  hr ln. angle: 0083.13368
Hr: 07.0  hr ln. angle: 0068.84415
Hr: 08.0  hr ln. angle: 0046.95435

Hr: 09.0  hr ln. angle: 0031.17341
Hr: 10.0  hr ln. angle: 0019.49255
Hr: 11.0  hr ln. angle: 0010.08288
----- noon -----
Hr: 12.0  hr ln. angle: 0001.73147
----- noon -----
Hr: 13.0  hr ln. angle: 0006.44690
Hr: 14.0  hr ln. angle: 0015.28246
Hr: 15.0  hr ln. angle: 0025.80996

Hr: 16.0  hr ln. angle: 0039.57639
Hr: 17.0  hr ln. angle: 0058.69652
Hr: 18.0  hr ln. angle: 0083.11535
ENTER to exit
    
```

NOTES:

If the compile, link, or execute brings up a Win32 error message, exit the system and restart KOBOL. Repeat the process and everything is usually corrected.

The ATAN function is good to about 45 degrees if the simple series for ATAN is used, the more complete series is good to 45 degrees if about 6 terms are used, and to over 70 degrees if about 70 terms are used. A loop of 500 iterations has the result good to 84 degrees when using larger resolution working storage: `pic 9999.999999999999999999`

The simplest ATAN series only works to 45 degrees, it is:

$$\text{atan} = x - (x**3)/3 + (x**5)/5 - (x**7)/7 + . . . \quad (x \leq 1)$$

The best series is:

$$\text{atan} = \left(\frac{x}{(1+x*x)} \right) * \left(1 + \frac{(2/3)*(x*x/(1+x*x))}{((2*4)/(3*5))*(x*x/(1+x*x))*(x*x/(1+x*x))} \right) \dots$$

and is an ATAN series good for x of any value. This works for angles greater than 45 degrees, as noted above.

While the series shown above may not be used much in the normal course of life, they were what the author used back in the IBM 360 and early 370 days when developing font rotation logic using the floating point feature of the computer.

COBOL sundial programming notes purely for legacy reasons

```
PROCEDURE DIVISION.
*****
* The main program itself
*****
    set lat to 0
    display '[COBOL-hDial-series.cob]'
    display 'function ATAN is approximate, and for angles' .
    display 'over 84 degrees valid to about 1 degree.' .
    display 'Enter a latitude' .
    accept lat
    set lng to 0
    display 'Enter a longitude' .
    accept lng
    set ref to 0
    display 'Enter reference longitude' .
    accept ref
    display ' '
    display 'Latitude, longitude, reference'
    display 'LAT: ', lat, ' LNG: ', lng, ' REF: ', ref
    display ' '
    compute corrmns = ( lng - ref ) * 4 .
    display 'correction in minutes: ', corrmns
    display ' '
    set hrbgn to 6
    set hrnow to 6
    set hrend to 18
    perform 13 times
* -----
* hrangle = atan ( sin(lat) * tan(hour) )
* -----
    if hrnow = 9 or hrnow = 16 then
        display ' '
    end-if

    if hrnow = 12 then
        display '----- noon -----'
    end-if

* get the sun's hour angle
    compute a = 12 - hrnow
    compute b = corrmns / 60
    if hrnow < 13 then
        compute angle = 15 * ( a + b )
    else
        compute angle = 15 * ( a - b )
    end-if

    set degAngle to angle
    perform TAN
* NOTE: TAN will clobber sinAngle and cosAngle - so do TAN before COS and SIN

    set degAngle to lat
    perform SIN

    compute angle = sinAngle * tanAngle
    set tanangle to angle
    perform ATAN
    compute degangle = radangle * 360 / ( 2 * 3.1416 )
    set temp to degangle
    display 'Hr: ', hrnow, ' hrln.angle: ', temp

    if hrnow = 12 then
        display '----- noon -----'
    end-if

    add 1 to hrnow
    end-perform
    display "ENTER to exit"
    accept angle
    exit.
stop run .
```

COBOL sundial programming notes purely for legacy reasons

```

*****
* sub routine for trigonometric sin
* INPUT:      degAngle
* RESULT:     sinAngle
*****
SIN.
*   sin = x - (x**3)/3! + (x**5)/5! - (x**7)/7! + . . .
  compute radAngle = degAngle*2*3.1416 / 360
  set x to radAngle
*   compute sinangle = x - x**3/3! + x**5/5! - x**7/7!
  compute a = x
  compute b = x*x*x/(3*2)
  compute c = x*x*x*x*x / (5*4*3*2)
  compute d = x*x*x*x*x*x*x/(7*6*5*4*3*2*1)
  compute e = x*x*x*x*x*x*x*x*x/(9*8*7*6*5*4*3*2*1)
  compute sinAngle = a - b + c - d + e
  exit .

*****
* sub routine for trigonometric cos
* INPUT:      degAngle
* RESULT:     cosAngle
*****
COS.
*   cos = 1 - (x**2)/2! + (x**4)/4! - (x**6)/6! + . . .
  compute radAngle = degAngle*2*3.1416 / 360
  set x to radAngle
*   compute sinangle = x - x**3/3! + x**5/5! - x**7/7!
  compute a = 1
  compute b = x*x/(2)
  compute c = x*x*x*x / (4*3*2)
  compute d = x*x*x*x*x*x/(6*5*4*3*2*1)
  compute e = x*x*x*x*x*x*x*x*x/(8*7*6*5*4*3*2*1)
  compute cosAngle = a - b + c - d + e
  exit .

*****
* sub routine for trigonometric tan
* INPUT:      degAngle
* RESULT:     tanAngle
* NOTE:      TAN clobbers sinAngle and tanAngle
*****
TAN.
*   tan = x + (x**3)/3 + 2*(x**5)/15 + . . .
  compute radAngle = degAngle*2*3.1416 / 360
  set x to radAngle
  perform SIN
  perform COS
  compute tanAngle = sinangle/cosangle
  exit .

*****
* sub routine for trigonometric atan
* INPUT:      tanAngle
* RESULT:     radAngle
*****
ATAN.
*   This series works for angles greater than 45, an error creeps in over 45
*   degrees that increases with angle when only 6 terms are used, the error
*   improves and is good up to 70 degrees with some 67 terms. At 500 terms
*   this is good to 84 degrees, and more terms and more precision on the variables
*   doesn't help much. Cheating is done after 84 degrees.
*
*   atan = ( x/(1+x*x) ) * ( 1 + (2/3)*(x*x/(1+x*x)) +
*   ((2*4)/(3*5))*(x*x/(1+x*x))*(x*x/(1+x*x)) ...
*   or if v = x*x/(1+x*x) and
*   and w = x/(1+x*x) then
*   atan = ( w ) * ( 1 + (2/3)*v + ((2*4)/(3*5))*v*v +
*   ((2*4*6)/(3*5*7))*v*v*v + ...
*

```


COBOL sundial programming notes purely for legacy reasons

```
set x to tanangle

*   angle--->radians--->tan
*   84      1.4661  9.5144
*   85      1.4835  11.4301
*   86      1.5010  14.3007
*   87      1.5184  19.0811
*   88      1.5359  28.6363
*   89      1.5533  57.2900
*   90      1.5708  16324552277619100.0000
*
* NOTE: The sequence of the following code is critical.
*
  if tanangle <= 9.5147 then
    compute v      =      (x*x) / (1+x*x)
    compute w      =      x / (1+x*x)
    compute a      =      1
*   b is previous set
    compute b      =      (2/3)*v
*   c is sum of all b sets
    compute c      =      b
*   k1 and k2 are multipliers used
    compute k1     =      4
    compute k2     =      5

    perform 500 times
    compute b      =      b*(k1/k2)*v
    compute c      =      b + c
    add 2 to k1
    add 2 to k2
    end-perform

    compute x      =      w * (1+c)
    set radAngle to x
  end-if

  if tanangle > 57.299 then
    compute radangle = 90*2*3.1416/360
  end-if
  if tanangle > 9.5148 and tanangle < 57.31 then
    compute radangle = 89*2*3.1416/360
  end-if
  if tanangle > 9.5148 and tanangle < 28.64 then
    compute radangle = 88*2*3.1416/360
  end-if
  if tanangle > 9.5148 and tanangle < 19.09 then
    compute radangle = 87*2*3.1416/360
  end-if
  if tanangle > 9.5148 and tanangle < 14.31 then
    compute radangle = 86*2*3.1416/360
  end-if
  if tanangle > 9.5148 and tanangle < 11.44 then
    compute radangle = 85*2*3.1416/360
  end-if
  exit .

* END *
```

COBOL sundial programming notes purely for legacy reasons

COBOL h-dial using functions

IDENTIFICATION DIVISION.
 PROGRAM-ID. COBOL-hDial-functions .
 DATE-WRITTEN. Friday Dec 12, 2008.
 DATE-COMPILED.

ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. Windows.
 OBJECT-COMPUTER. Windows.

DATA DIVISION.

WORKING-STORAGE SECTION.

* work variables for the main program itself

```

77 angle          pic 9999.99999 .
77 lat           pic 9999.99999 .
77 lng          pic 9999.99999 .
77 ref          pic 9999.99999 .
77 corrmins     pic 9999.99999 .
  
```

```

77 hrbgn        pic 99.9 .
77 hrnow       pic 99.9 .
77 hrend       pic 99.9 .
  
```

```

77 hrangle     pic 9999.99 .
  
```

* work variables for sub routines used by PERFORM of SIN, COS, and TAN, etc

```

77 a           pic 9999.99999 .
77 b           pic 9999.99999 .
  
```

```

77 degAngle    pic 9999.99999 .
77 radAngle    pic 9999.99999 .
  
```

```

77 sinAngle    pic 9999.99999 .
77 cosAngle    pic 9999.99999 .
77 tanAngle    pic 9999.99999 .
  
```

```

77 atanAngle   pic 9999.99 .
  
```

* -----

* test data for latitude - 32.75 longitude 108.2/105.0

* -----

*	Hour line angles for			Horizontal hour line	
*	TIME OF DAY			angles with long corr	
*	am	pm		am	pm
*	12.00	12.00	0.00	1.73	-1.73
*	11.00	1.00	8.25	10.09	6.45
*	10.00	2.00	17.35	19.49	15.28
*	9.00	3.00	28.41	31.18	25.81
*	8.00	4.00	43.14	46.96	39.58
*	7.00	5.00	63.65	68.88	58.71
*	6.00	6.00	90.00	-84.10	84.10
*			These have no	These have	
*			longitude correction	longitude correction	

PROCEDURE DIVISION.

* The main program itself

```

set lat to 0
display '[COBOL-hDial-functions.cob]'
display 'angles seem ok using inbuilt functions' .
display 'Enter a latitude' .
accept lat
  
```

```

set lng to 0
  
```

COBOL sundial programming notes purely for legacy reasons

```
display 'Enter a longitude'      .
accept lng

set ref to 0
display 'Enter reference longitude' .
accept ref

display ' '
display 'Latitude, longitude, reference'
display 'LAT: ', lat, ' ' LNG: ', lng, ' ' REF: ', ref
display ' '

compute corrmns = ( lng - ref ) * 4 .
display 'correction in minutes: ', corrmns
display ' '

set hrbgn to 6
set hrnow to 6
set hrend to 18

perform 13 times
* -----
* hrangle = atan ( sin(lat) * tan(hour) )
* -----
  if hrnow = 9 or hrnow = 16 then
    display ' '
  end-if

  if hrnow = 12 then
    display '----- noon -----'
  end-if
*
  get the sun's hour angle
  compute a = 12 - hrnow
  compute b = corrmns / 60
  if hrnow < 13 then
    compute angle = 15 * ( a + b )
  else
    compute angle = 15 * ( a - b )
  end-if

  compute radangle = angle*2*3.1416 / 360
  compute angle = radangle
  compute tanAngle = function TAN(angle)

  compute sinAngle = function sin(lat)

  compute angle = sinAngle * tanAngle
  set tanangle to angle
  set radangle to function atan(tanangle)
  compute degangle = radangle*360/ (2*3.1416)

  display 'Hr: ', hrnow, ' hrln.angle: ',degangle

  if hrnow = 12 then
    display '----- noon -----'
  end-if

  add 1 to hrnow
end-perform

display "ENTER to exit"
accept angle
exit.
stop run .
* END *
```