

Abacus

Lesson 1

By Dr. Michael J. Bisconti
and
ABPROTO (Abacus Programming Team)
at
Chicago Christian University

Abacus Source Code

The time has come for us to begin to unveil the mysteries of Abacus, our computer brain. As we have stated previously, Abacus is a *software* brain. The importance of this is the *portability* of this technology to *any computer anywhere in the world*.

Below you will find the source code for Abacus. At first glance, Abacus will appear to be a very simple program. It isn't. However, it is a program that is *understandable by anyone with just basic programming skills*. This is possible because we have converted the VectorSineM (proprietary language) code into a form of the BASIC programming language. To be exact, the Open Source FreeBASIC language. You can find tons of information about FreeBASIC on the internet. Start with <http://sourceforge.net/projects/fbc/?source=directory>.

Why are we sharing our secrets with you and the rest of the world? Because you have the right to know how we arrive at our conclusions; in particular, our conclusions about what is the correct original language and English language text of the Bible.

This document is the next step in a lengthy journey.

/******

*

* Name: abacus.bas

*

* Synopsis: Abacus artificial intelligence written in VectorSineM converted to FreeBASIC.

*

* Description: This is the Abacus program rewritten so that it can be understood

* by any programmer and run on any home computer system.

* Note that this code is written for the Open Source FreeBASIC compiler.

* In addition, we have modified the compiler and will be providing the

* modified compiler code at some point in the future.

* One of our innovations is the "EvokePool" procedure call.

* (EvokePool is a "call caller." It allows us to program much more quickly.)

* EvokePool uses dynamic encryption. This is a precaution to safeguard

* against misuse of Abacus.

*

* Copyright 1980 - 2015 Dr. Michael J. Bisconti

*

* Credits: ABPROTO (Abacus Programming Team) at Chicago Christian University

*

*****/

#IncludePool(#Include "000000.bi",1,1000233)

'Object enumeration

Enum Depth

 Evoke depth.dll

End Enum

SetDim variable.dll As collection.dll

'Object type

Type objtype

 objectset.dll

End Type

Set the default parameters.

Constructor objtype ()

 parcon = parameters.dll

End Constructor

'Sets and returns object type.

Property objtype.OType (e As eobject)

 parcon = e

```
End Property

Function InitLevel () As gamemaster
    EvokePool(Evoke function0000001.dll,1,877)
End Function

'Set up textual filters
Sub SetUpTextualFilters
    EvokePool(Evoke Filter0000001.dll,1,10099999)
End Sub

'Get things done.
Sub IntegrateTextualFilters
    PoolEvoke(Evoke Integrator0000001.dll,1,2000000)
End Sub

Sub ProcessText
    PoolEvoke(Evoke processor0000001.dll,1,1000233)
End Sub

Sub GenerateResults
    PoolEvoke(Evoke generator0000001.dll,1,1000233)
End Sub

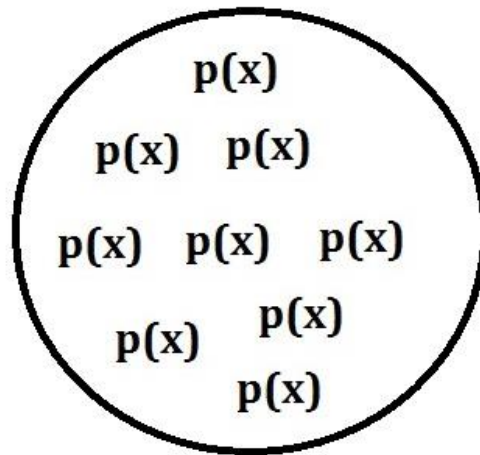
'End program
Do
    Cls
    Print
    Print "Do you want to play again? [Y/N]"
    Do
        skey = InKey
        skey = UCase(skey)
    Loop Until (skey = "Y") Or skey = "N"
    If skey = "N" Then
        gameover = TRUE
    Endif
Loop Until gameover = TRUE
Print "Game over. You won. Have a nice day."
```

Lesson 1

InvokePool Versus PoolInvoke

In this lesson, we will talk about the difference between InvokePool and PoolInvoke. InvokePool calls a pool (collection) of “programs” (note quotes). PoolInvoke calls a pool of pools (collection of collections) of “programs” (note quotes).

InvokePool



PoolInvoke

