# wineforth_502
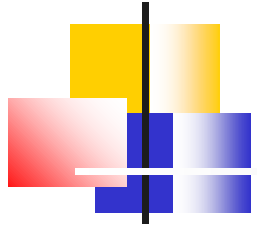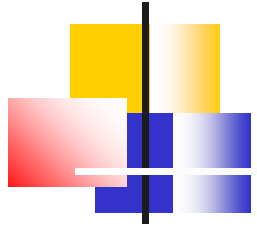
## Silicon Valley
## Forth Interest Group
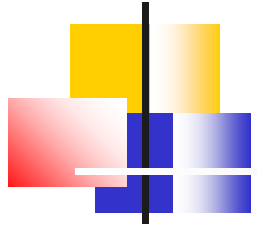
**Chen-Hanson Ting**
**August 28, 2021**

# wineForth

- **It should be pronounced win-e-Forth**
- **It could also be pronounced wine-Forth**

# ceForth

- **ceForth was eForth written in C.**
- **ceForth_21 in Cygwin cpp used a dictionary meta-complied by weForth.**
- **ceForth_33 in Visual Studio 2017 had a meta-compiler to build Forth dictionary in runtime.**

# ceforth_403
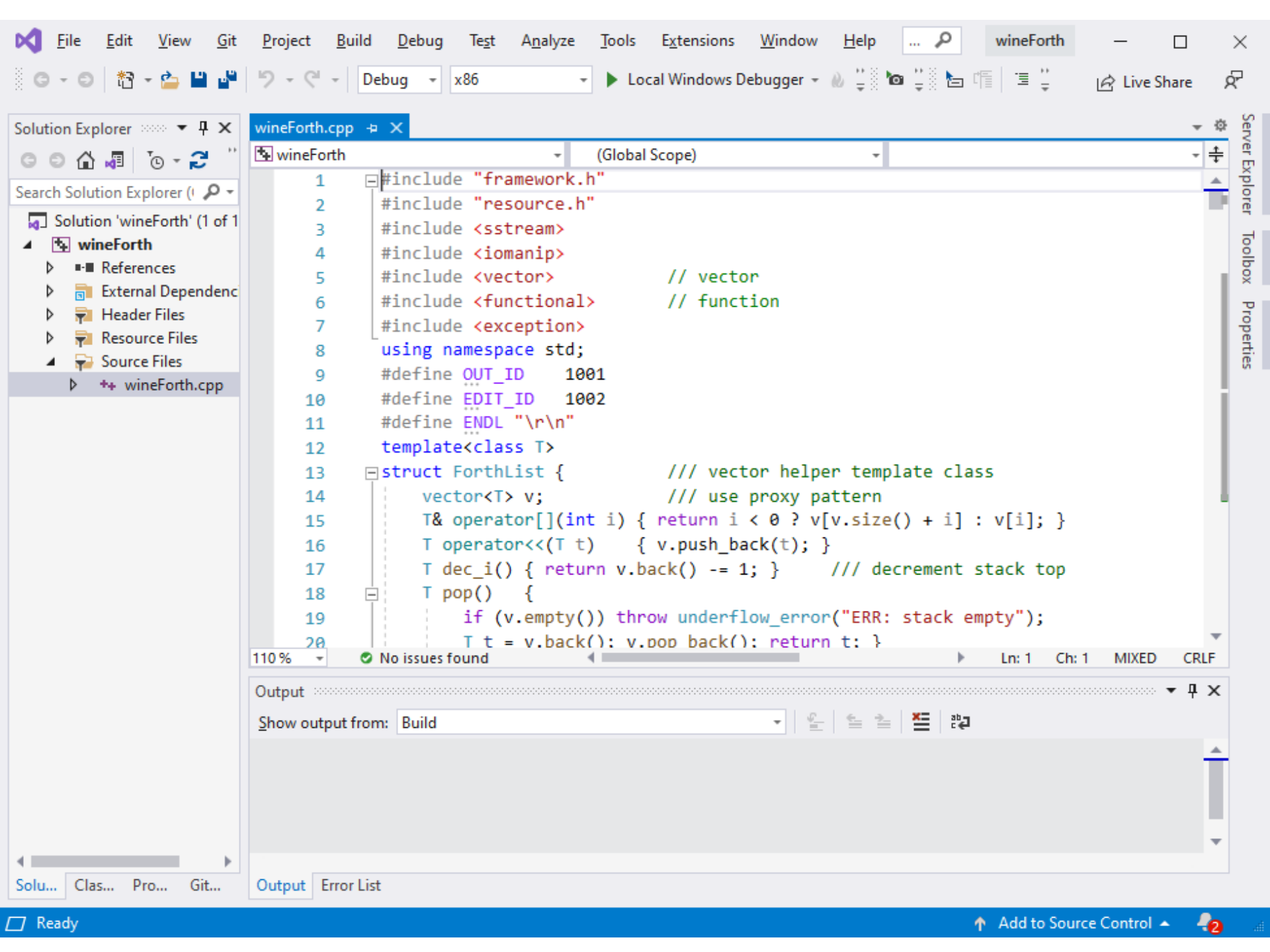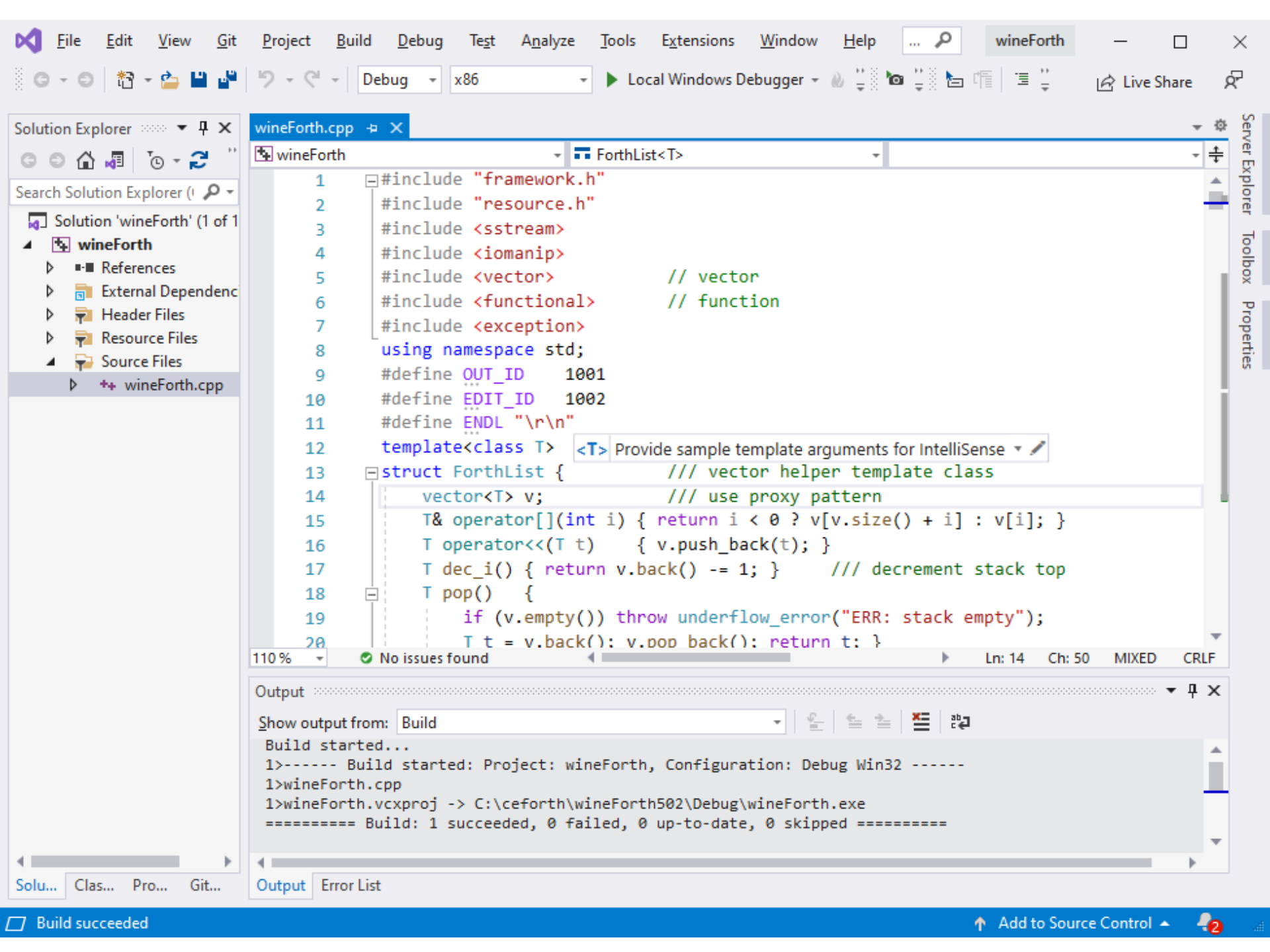
- **It was written by Chochain Lee as a C++ console application in Visual Studio 2019 Community.**
- **It runs in a console window like an old-fashion Forth program with a DOS terminal.**
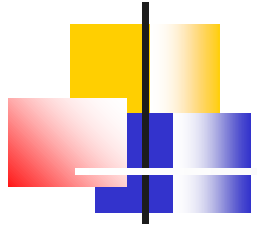
# wineforth_502

- **Chochain changes ceforth_403 to wineforth_502.**
- **A simple Windows interface allows text in an input window to be evaluated and output characters are sent to an output window.**

Debug  x86  ▶ Local Windows Debugger  Live Share

Search Solution Explorer (

Solution 'wineForth' (1 of 1
- wineForth
  - References
  - External Dependenc
  - Header Files
  - Resource Files
  - Source Files
    - wineForth.cpp

wineForth.cpp

wineForth                                    (Global Scope)

```cpp
1    #include "framework.h"
2    #include "resource.h"
3    #include <sstream>
4    #include <iomanip>
5    #include <vector>          // vector
6    #include <functional>      // function
7    #include <exception>
8    using namespace std;
9    #define OUT_ID    1001
10   #define EDIT_ID   1002
11   #define ENDL "\r\n"
12   template<class T>
13   struct ForthList {          /// vector helper template class
14       vector<T> v;             /// use proxy pattern
15       T& operator[](int i) { return i < 0 ? v[v.size() + i] : v[i]; }
16       T operator<<(T t)    { v.push_back(t); }
17       T dec_i() { return v.back() -= 1; }      /// decrement stack top
18       T pop()   {
19           if (v.empty()) throw underflow_error("ERR: stack empty");
20           T t = v.back(); v.pop_back(); return t; }
```

110 %        ✓ No issues found                                    Ln: 1    Ch: 1    MIXED    CRLF

Output

Show output from: Build

Output    Error List

Ready                                                                    ↑ Add to Source Control

Debug   ▾   x86   ▾   ▶ Local Windows Debugger ▾      🖉 Live Share   ⌗

Solution Explorer                 ⚓ ✕

Search Solution Explorer (       🔍 ▾

🔷 Solution 'wineForth' (1 of 1
◢ 🔷 wineForth
  ▷ ■▪ References
  ▷ 🔲 External Dependenc
  ▷ 🔲 Header Files
  ▷ 🔲 Resource Files
  ◢ 🔲 Source Files
    ▷ ✚✚ wineForth.cpp

wineForth.cpp  ⯐ ✕

🔷 wineForth                    ▾      🔲 ForthList<T>                    ▾

```cpp
 1    ⊟#include "framework.h"
 2     #include "resource.h"
 3     #include <sstream>
 4     #include <iomanip>
 5     #include <vector>          // vector
 6     #include <functional>      // function
 7     #include <exception>
 8     using namespace std;
 9     #define OUT_ID     1001
10     #define EDIT_ID    1002
11     #define ENDL "\r\n"
12     template<class T>    <T> Provide sample template arguments for IntelliSense ▾ 🖉
13    ⊟struct ForthList {          /// vector helper template class
14         vector<T> v;            /// use proxy pattern
15         T& operator[](int i) { return i < 0 ? v[v.size() + i] : v[i]; }
16         T operator<<(T t)    { v.push_back(t); }
17         T dec_i() { return v.back() -= 1; }     /// decrement stack top
18    ⊟    T pop()    {
19             if (v.empty()) throw underflow_error("ERR: stack empty");
20             T t = v.back(); v.pop_back(); return t; }
```

110 %  ▾     ✓ No issues found                                    Ln: 14   Ch: 50   MIXED   CRLF

Output                                                              ▾ �📌 ✕

Show output from:  Build                    ▾      ⭰ ⭰ ⭲ ✖ 🔤

```
 Build started...
 1>------ Build started: Project: wineForth, Configuration: Debug Win32 ------
 1>wineForth.cpp
 1>wineForth.vcxproj -> C:\ceforth\wineForth502\Debug\wineForth.exe
 ========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Output   Error List

▱ Build succeeded                                    ⬆ Add to Source Control ▴    🔔2

wineForth v502

words

```
words
dup 0 over 1 2dup 2 2over 3 4dup 4 swap 5 rot 6 -rot 7
2swap 8 pick 9 drop 10 nip 11 2drop 12 >r 13 r> 14 r@ 15
+ 16 - 17 * 18 / 19 mod 20 */ 21 */mod 22 and 23
or 24 xor 25 negate 26 abs 27 0=  28 0< 29 0> 30 = 31
> 32 < 33 <> 34 >= 35 <= 36 base@ 37 base! 38 hex 39
decimal 40 cr 41 . 42 .r 43 u.r 44 .f 45 key 46 emit 47
space 48 spaces 49 dotstr 50 dolit 51 dovar 52 [ 53 ] 54 ( 55*
.( 56* \ 57 $" 58 ." 59* bran 60* if 61* else 62* then 63*
loop 64 begin 65* while 66* repeat 67* again 68* until 69* cycle 70
for 71*
aft 72* next 73* exit 74 exec 75 : 76 ; 77* variable 78 constant 79
@ 80 ! 81 +! 82 ? 83 array@ 84 array! 85 allot 86 , 87
create 88 does 89 [to] 90 to 91 is 92 here 93 words 94 .s 95
' 96 see 97 forget 98 boot 99  <-1> ok
```

# Lesons502.txt

```
cr .( example 1.         the universal greeting )

decimal

: hello cr ." hello, world!" ;

cr .( example 2.        the big f )

: bar    cr ." *****" ;
: post   cr ." *      " ;
: f       bar post bar post post post ;

( type 'f' and a return on your keyboard, and you will see a large
f character displayed on the screen )


cr .( example 3.        fig, forth interest group )

: center cr ."    *   " ;
: sides  cr ." *    *" ;
: triad1 cr ." * * *" ;
: triad2 cr ." **   *" ;
: triad3 cr ." *   **" ;
: triad4 cr ."   ***  " ;
: quart  cr ." ** **" ;
: right  cr ." * ***" ;
: bigt  bar center center center center center center ;
: bigi  center center center center center center center ;
: bign  sides triad2 triad2 triad1 triad3 triad2 sides ;
: bigg  triad4 sides post right triad1 sides triad4 ;
```

```
words
dup 0 over 1 2dup 2 2over 3 4dup 4 swap 5 rot 6 -rot 7
2swap 8 pick 9 drop 10 nip 11 2drop 12 >r 13 r> 14 r@ 15
+ 16 - 17 * 18 / 19 mod 20 */ 21 */mod 22 and 23
or 24 xor 25 negate 26 abs 27 0=  28 0< 29 0> 30 = 31
> 32 < 33 <> 34 >= 35 <= 36 base@ 37 base! 38 hex 39
decimal 40 cr 41 . 42 .r 43 u.r 44 .f 45 key 46 emit 47
space 48 spaces 49 dotstr 50 dolit 51 dovar 52 [ 53 ] 54 ( 55*
.( 56* \ 57 $" 58 ." 59* bran 60* if 61* else 62* then 63*
loop 64 begin 65* while 66* repeat 67* again 68* until 69* cycle 70
for 71*
aft 72* next 73* exit 74 exec 75 : 76 ; 77* variable 78 constant 79
@ 80 ! 81 +! 82 ? 83 array@ 84 array! 85 allot 86 , 87
create 88 does 89 [to] 90 to 91 is 92 here 93 words 94 .s 95
' 96 see 97 forget 98 boot 99  <-1> ok
```

```
cr .( example 1.         the universal greeting )

decimal

: hello cr ." hello, world!" ;

cr .( example 2.         the big f )

: bar   cr ." *****" ;
: post  cr ." *    " ;
: f     bar post bar post post post ;

( type 'f' and a return on your keyboard, and you will
see a large
f character displayed on the screen )


cr .( example 3.         fig, forth interest group )

: center cr ."   *  " ;
: sides  cr ." *   *" ;
: triad1 cr ." * * *" ;
: triad2 cr ." ** *" ;
: triad3 cr ." *  **" ;
: triad4 cr ."  *** " ;
: quart  cr ." ** **" ;
: right  cr ." * ***" ;
: bigt  bar center center center center center center ;
: bigi  center center center center center center center
;
: bign  sides triad2 triad2 triad1 triad3 triad2 sides ;
: bigg  triad4 sides post right triad1 sides triad4 ;
: fig  f bigi bigg ;


cr .( example 4.         repeated patterns )
(
for    [ index -- ]             set up loop given the
index.
next   [ -- ]                   decrement index by 1.  if
```

```
example 6.      help
example 7.      money exchange
example 8.      temperature conversion
example 9.      weather reporting.
example 10.     print the multiplication table

          1    2    3    4    5    6    7    8    9   10   11   12
  1       1    2    3    4    5    6    7    8    9   10   11   12
  2       2    4    6    8   10   12   14   16   18   20   22   24
  3       3    6    9   12   15   18   21   24   27   30   33   36
  4       4    8   12   16   20   24   28   32   36   40   44   48
  5       5   10   15   20   25   30   35   40   45   50   55   60
  6       6   12   18   24   30   36   42   48   54   60   66   72
  7       7   14   21   28   35   42   49   56   63   70   77   84
  8       8   16   24   32   40   48   56   64   72   80   88   96
  9       9   18   27   36   45   54   63   72   81   90   99  108
 10      10   20   30   40   50   60   70   80   90  100  110  120
 11      11   22   33   44   55   66   77   88   99  110  121  132
 12      12   24   36   48   60   72   84   96  108  120  132  144

example 11.     calendars
*****************************************************************
    sun      mon      tue      wed      thu      fri      sat
*****************************************************************
                                                             1
    2        3        4        5        6        7        8
    9       10       11       12       13       14       15
   16       17       18       19       20       21       22
   23       24       25       26       27       28       29
   30       31
*****************************************************************

example 12.     sines and cosines
example 13.     square root
example 14.     radix for number conversions
example 15.     ascii character table
example 16.     random numbers
example 17.     guess a number  <-1> ok
```

```
        16        17        18        19        20        21        22
        23        24        25        26        27        28        29
        30        31
****************************************************************

 example 12.     sines and cosines
 example 13.     square root
 example 14.     radix for number conversions
 example 15.     ascii character table
 example 16.     random numbers
 example 17.     guess a number  <-1> ok
words
dup 0 over 1 2dup 2 2over 3 4dup 4 swap 5 rot 6 -rot 7
2swap 8 pick 9 drop 10 nip 11 2drop 12 >r 13 r> 14 r@ 15
+ 16 - 17 * 18 / 19 mod 20 */ 21 */mod 22 and 23
or 24 xor 25 negate 26 abs 27 0=  28 0< 29 0> 30 = 31
> 32 < 33 <> 34 >= 35 <= 36 base@ 37 base! 38 hex 39
decimal 40 cr 41 . 42 .r 43 u.r 44 .f 45 key 46 emit 47
space 48 spaces 49 dotstr 50 dolit 51 dovar 52 [ 53 ] 54 ( 55*
.( 56* \ 57 $" 58 ." 59* bran 60* if 61* else 62* then 63*
loop 64 begin 65* while 66* repeat 67* again 68* until 69* cycle 70
for 71*
aft 72* next 73* exit 74 exec 75 : 76 ; 77* variable 78 constant 79
@ 80 ! 81 +! 82 ? 83 array@ 84 array! 85 allot 86 , 87
create 88 does 89 [to] 90 to 91 is 92 here 93 words 94 .s 95
' 96 see 97 forget 98 boot 99 hello 100 bar 101 post 102 f 103
center 104 sides 105 triad1 106 triad2 107 triad3 108 triad4 109 quart
110 right 111
bigt 112 bigi 113 bign 114 bigg 115 fig 116 width 117 asterisks 118
rectangle 119
parallelogram 120 triangle 121 the 122 that 123 this 124 jack 125
summary 126 flaw 127
mummery 128 k 129 haze 130 phrase 131 bluff 132 stuff 133 theory 134
button 135
child 136 cybernatics 137 hiding 138 lay 139 based 140 saved 141 cloak
142 thick 143
hung 144 cover 145 make 146 pushed 147 without 148 rest 149 cloaked
150 THEORY 151
question 152 help 153 sex 154 too 155 much 156 little 157 health 158
job 159
```
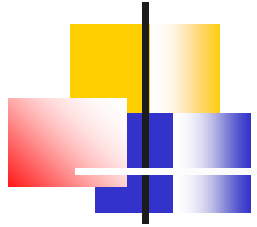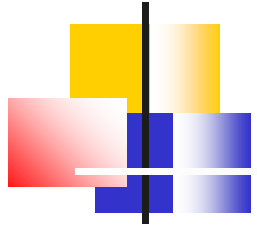
# Forth Virtual Machine

- A dictionary
- An outer interpreter
- An inner interpreter
- A return stack
- A parameter stack
- An IO device

# Forth Virtual Machine

- **Forth words are objects.**
- **Primitive words, outer interpreter, and inner interpreter can be written in object-oriented languages.**
- **jeforth, ooeforth, ceforth, and wineforth are poetry.**

# ESP32eforth

- **A goal of ceforth_403 was to replace the ESP32eforth running in ESP32 chips.**
- **ESP32eforth was pathetic.**
- **Chochain had already ported eforth_403 to ESP8266 and ESP32.**
- **New esp32forth is beautiful.**
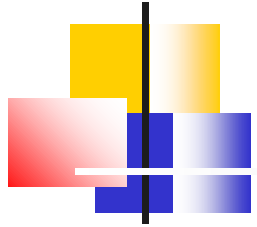
# **Windows Forth Saga**

- **Two quarrelsome frogs lived in a deep well.**
- **They heatedly debated how big the sky was.**
- **With objective observation and accurate measurements, they finally agreed that the sky was circular and 5 feet in diameter.**
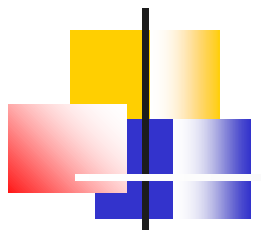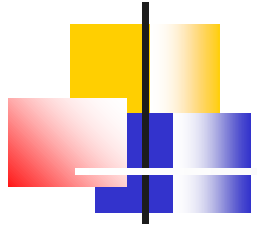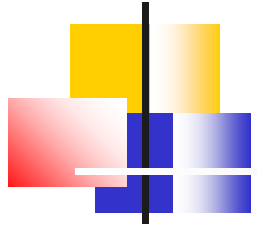
# Windows Forth Saga

- **Chochain and I were these frogs in a well called Forth.**
- **We argued how big the sky was.**
- **After all the work, we decided the sky was 1080 pixels wide and 680 pixels high.**
- **That's Windows.**

# Thank You!