CS301
Programming Assignment #3
Thomas Crow
10/28/22

1) ADA

⚙ Execute | 💬 Beautify | ⤳ Share | Source Code | ⓘ Help

```ada
1   --CS301
2   --Thomas Crow
3   --
4   -- Requirements:
5   -- function f()  {
6   -- write('I have been evaluated');
7   -- return(1);  }
8   -- main()
9   -- {
10  -- int i=1;
11  -- if (i == 0 && f() ) then write ('true') else write ('false')
12  -- }
13
14  with Ada.Text_IO;
15  use Ada.Text_IO;
16
17  procedure shortcircuit is
18
19      function f
20          return Integer is
21      begin
22          put_line("I have been evaluated");
23          return 1;
24      end f;
25
26      i : integer;
27
28  begin
29      i := 1;
30      if (i = 0) and (f = 1)
31      then
32          put_line("True");
33      else
34          put_line("False");
35      end if;
36  end shortcircuit;
37
38
```

>_ Terminal

```
gcc -c hello.adb
hello.adb:17:11: warning: file name does not match unit name, should be "shortcircuit.adb"
gnatbind -x hello.ali
gnatlink hello.ali -o hello
I have been evaluated
False
```

1) Pascal

```pascal
(* {
CS301
Thomas Crow

Requirements:
function f()
{
    write('I have been evaluated');
    return(1);
}
main()
{
int i=1;
if (i == 0 && f() ) then write ('true') else write ('false')
}
*)


program ShortCircuit;
var
 i : integer;

  function f(): integer;

  begin
  writeln('I have been evaluated');
  f := 1;
  end;


begin
  i := 1;
  if i = 0 and f()
  then
    writeln('true')
  else
    writeln('false');
end.
```

```
Free Pascal Compiler version 3.0.4+dfsg-23 [2019/11/25] for x86_64
Copyright (c) 1993-2017 by Florian Klaempfl and others
Target OS: Linux for x86-64
Compiling main.pas
Linking a.out
/usr/bin/ld.bfd: warning: link.res contains output sections; did you forget -T?
39 lines compiled, 0.1 sec
false


...Program finished with exit code 0
Press ENTER to exit console.
```

1) Go

Execute | Beautify | Share | Source Code | Help

```go
1   // CS301
2   // Thomas Crow
3   //
4   // Requirements:
5   // function f()
6   // {
7   //    write('I have been evaluated');
8   //    return(1);
9   // }
10  // main()
11  // {
12  // int i=1;
13  // if (i == 0 && f() ) then write ('true') else write ('false')
14  // }
15
16  package main
17
18  import "fmt"
19  func f () int {
20      fmt.Printf("I have been evaluated\n")
21      return 1
22  }
23
24  func main() {
25
26      var i int = 1
27      if (i == 0) && (f() == 1) {
28          fmt.Printf("true")
29      } else {
30          fmt.Printf("false")
31      }
32  }
```

Terminal

```
false
```

1) Fortran
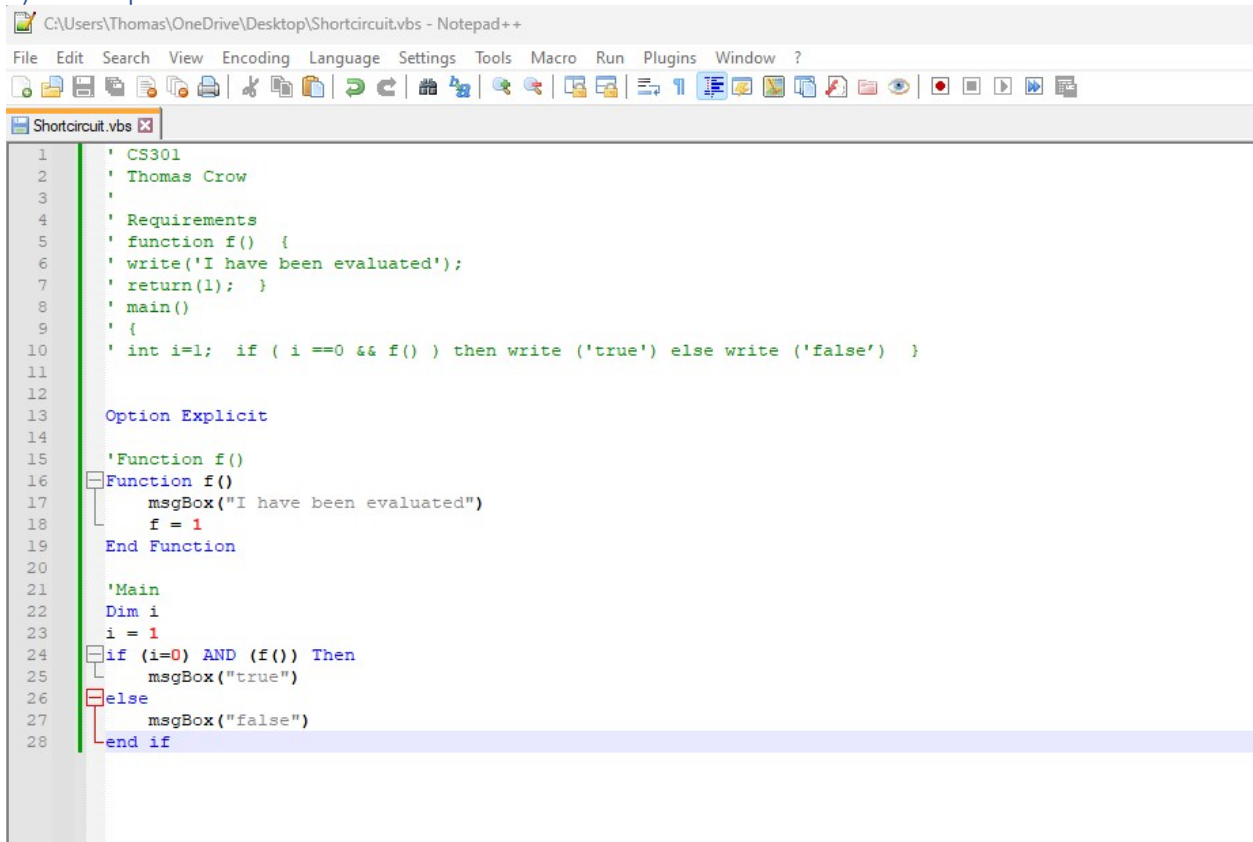
```fortran
1   !CS301
2   ! Thomas Crow
3   !
4   ! Requirements:
5   ! function f()
6 ⌄ ! {
7   !     write('I have been evaluated');
8   !     return(1);
9   ! }
10  ! main()
11⌄ ! {
12  ! int i=1;
13  ! if (i == 0 && f() ) then write ('true') else write ('false')
14  ! }
15
16
17
18  program main
19  integer :: i, j
20      i = 1
21      if ( i .eq. 0 .and. f() .eq. 1) THEN
22          Print *, "true"
23      else
24          Print *, "false"
25      end if
26
27  end program main
28
29
30  function f() result(x)
31      Print *, "I have been evaluated"
32      x = 1
33  end function f
34
```

Terminal

```
false
```

1) VBScript

File   Edit   Search   View   Encoding   Language   Settings   Tools   Macro   Run   Plugins   Window   ?

Shortcircuit.vbs

```vbscript
 1    ' CS301
 2    ' Thomas Crow
 3    '
 4    ' Requirements
 5    ' function f()  {
 6    ' write('I have been evaluated');
 7    ' return(1);  }
 8    ' main()
 9    ' {
10    ' int i=1;  if ( i ==0 && f() ) then write ('true') else write ('false')  }
11
12
13    Option Explicit
14
15    'Function f()
16    Function f()
17        msgBox("I have been evaluated")
18        f = 1
19    End Function
20
21    'Main
22    Dim i
23    i = 1
24    if (i=0) AND (f()) Then
25        msgBox("true")
26    else
27        msgBox("false")
28    end if
```

I have been evaluated

OK

false

OK
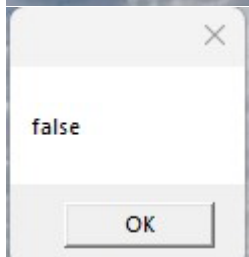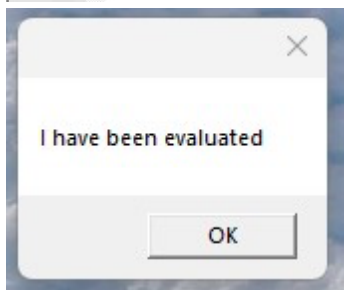
2) The languages Fortran, Pascal and Go used short circuit and did not evaluate the second conditional, which ran the function. Ada and VBScript did not and ran the second function.

3) While in our program, the second conditional was trivial in program execution, this is not always the case. If the second conditional had a significantly costly execution cost, skipping the second conditional could save considerable execution time.