SOURCE CODE EXAMPLES: FORTRAN

This appendix contains the code examples referred to in the main text. There is a version of the appendix for each language. This is the Fortran version.

#1 EXAMPLE FOR THE CALLING DIAGRAM

subroutine main()
  call function_one()
  call function_two()
end

subroutine function_one()
  call function_three()
end

subroutine function_two()
  call function_four()
  call function_one()
  call function_two()
end

subroutine function_three()
end

subroutine function_four()
  call function_three()
end

#2 SMALL EXAMPLE FOR HAND SIMULATION

integer v1, v2, excess

v1 = 10
v2 = v1 * 100
100 if (v2.le.500) then
  goto 200
endif
excess = v2 - 500
print *, 'Excess over 500 = ', v2 - 500
v1 = v1 - 1
v2 = v1 * 100
goto 100
200 ...
#3 EXAMPLE FOR SCOPE REGIONS

Fortran has only two levels of scope. Variables can be local to a subroutine or function (this applies to parameters as well) or they can accessible in a common block to any subprogram. Even then variables in a common block are only accessible if the declaration is repeated in the body of the subprogram.

```fortran
subroutine A()
    integer B
end

C

subroutine E()
    integer C
end
```

#4 LARGE EXAMPLE FOR HAND SIMULATION

```fortran
block data
    integer g1, g2
    common /global/ g1, g2
    data g1, g2 /100, 20/
end

C

subroutine f1(xx)
    integer xx
    integer w, res
    integer g1, g2
    common /global/ g1, g2
    w = 10
    w = w + g1
    res = w / xx
    print *, res
end

C

integer function f2()
    integer w
    integer g1, g2
    common /global/ g1, g2
    w = 5
    g1 = g2 + w
    f2 = w + 2
end

C

main program
    integer r, g1, g2
    common /global/ g1, g2
    integer f2
    r = f2()
    call f1(r)
    print *, g1
end
```
#5 TOP-DOWN DESIGN EXAMPLE

C The paint estimator
block data
  real paintc, height
  common /consts/ paintc, height
data paintc, height/12.5, 8.0/
  real totalc, length, width
  common /globals/ totalc, length, width, cover, area
end

C subroutine invals()
  real paintc, height
  common /consts/ paintc, height
  real totalc, length, width, cover, area
  common /globals/ totalc, length, width, cover, area
  print *, 'Type in length, width, coverage:'
  read *, length, width, cover
end

C subroutine calca()
  real paintc, height
  common /consts/ paintc, height
  real totalc, length, width, cover, area
  common /globals/ totalc, length, width, cover, area
  area = 2 * (length + width) * height + length * width
  print *, area
end

C subroutine calcca()
  real paintc, height
  common /consts/ paintc, height
  real totalc, length, width, cover, area
  common /globals/ totalc, length, width, cover, area
  print *, area, cover, paintc
  totalc = area / cover * paintc;
end

C subroutine calcc()
call calca()
call calcca()
end

C subroutine dispcc()
  real totalc, length, width, cover, area
  common /globals/ totalc, length, width, cover, area
  print *, 'Cost of painting a room is ', totalc
end
C#6  TOP-DOWN DESIGN EXAMPLE, REWORKED

C The Paint Estimator
C
  block data
    real paintc, height
    common /consts/ paintc, height
  data paintc, height /12.5, 8.0/
end

C subroutine invs(l, w, c)
  real l, w, c
C
  print *, 'Type in length, width, coverage:'
  read *, l, w, c
end

C real function calca(l, w)
  real l, w
C
    real paintc, height
    common /consts/ paintc, height
    calca = 2 * (l + w) * height + l * w
end

C real function calcca(a, c)
  real a, c
C
    real paintc, height
    common /consts/ paintc, height
    calcca = a / c * paintc;
end

real function calcc(l, w, c)
  real l, w, c, area
C
    real calca
    area = calca(l, w)
    calcc = calcca(area, c)
end
#7 THE MAIN PROGRAM FOR THE TOP-DOWN DESIGN EXAMPLE

C main program
  float costp, length, width, coverp;
  
  length = inputl()
  width = inputw()
  coverp = inputc()
  costp = calcc(length, width, coverage);
  call dispclength, width, coverage, costp);
  
#8 THE PAINT ESTIMATOR DESIGN IN CLASSES

Since Fortran does not have classes, we must simulate the encapsulation of data with named common areas within a block data subprogram, and the encapsulation of subprograms through naming conventions. The common areas are:

```fortran
block data RoomClass
  real width, length, height
  common /Room/ width, length, height
  save /Room/
  data height /8.0/
end

block data PaintClass
  real pcost, cover
  common /Paint/ pcost, cover
  save /Paint/
  data pcost /12.5/
end

block data EstClass
  real costp
  common /Est/ costp
end
```
#9 THE PAINT ESTIMATOR DESIGN WITH CLASS METHODS

The methods will be functions whose name begin with the first letter of the “class” name. There is no need for accessor methods or the constructor in this case, since we can use the variables in the common blocks directly. Initialization is done in the block data subprograms.

C Room methods
subroutine RInput()
   real width, length, height
   common /Room/ width, length, height
   print *, 'Type width and length: '
   read *, width, length
end

C Paint methods
subroutine PInput()
   real pcost, coverp
   common /Paint/ pcost, cover
   print *, 'Type coverage: '
   read *, cover
end

C
subroutine EDisp()
   real costp
   common /Est/ costp
   print *, 'Cost is ', costp
end

C
subroutine ERun()
defined below
end

#10 THE PAINT ESTIMATOR TOP LEVEL METHODS

subroutine EEstm()
   real warea, carea, width, length, height
   real costp, coverp
   common /Room/ width, length, height
   common /Paint/ pcost, coverp
   common /Est/ costp
   warea = 2 * (width + length) * height
   carea = width * length
   print *, warea, carea
   costp = (warea + carea) / coverp * pcost
end

C
subroutine ERun()
call RInput()
call PInput()
call EEstm()
call EDisp()
end
C    main program
    call ERun()
end

#11 SIMPLE EXAMPLE FOR DEBUGGING

    do 100 n = 1, 10
    total = total + num
    read *, num
    100 continue

#12 DEBUGGING WITH TRACE ADDED

    do 100 n = 1, 10
    print *, 'Trace total: ', total
    total = total + num
    read *, num
    100 continue

#13 ADDING AN ASSERTION

Fortran does not have a built-in assertion mechanism. See Example #14.

#14 AN ASSERTION IN STRAIGHT CODE

    if (x .lt. 0 .or. x .gt. 9) then
      print *, 'Error: x is out of range ', x
      stop
    endif

#15 A DEBUGGING EXAMPLE

    character*1 c, lastc
    character*80 line
    integer totalsies, totaleis

*    totalsies = 0
*    totaleis = 0
*    lastc = char(0)
*    print *, 'Type any number of lines followed by EOF'
*    read (unit = *, fmt = 101, end = 1000) line
100 format (A80)
101 index = 1
200 c = line(index:index)
    index = index + 1
    if (index .gt. 80) goto 100
    if (c .ge. 'A' .and. c .le. 'Z') then
      c = char(ichar(c) + 32)
    endif
    if (c .eq. 'i' .and. lastc .eq. 'e') then
      totaleis = totaleis + 1
    else if (c .eq. 'e' .and. lastc .eq. 'i') then
totalies = totalies + 1
else
  lastc = c
endif
goto 200
*
1000 print *, ‘the number of IEs is’, totalies
print *, ‘and the number of EIs is’, totaleis
end

#16 THE EXAMPLE WITH TRACE ADDED

character*1 c, lastc
character*80 line
integer totalies, totaleis
*
  totalies = 0
  totaleis = 0
  lastc = char(0)
*
  print *, ‘Type any number of lines followed by EOF’
*
100 read (unit = *, fmt = 101, end = 1000) line
101 format (A80)
  index = 1
200 c = line(index:index)
  index = index + 1
if (index .gt. 80) goto 100
if (c .ge. ‘A’ .and. c .le. ‘Z’) then
  c = char(ichar(c) + 32)
endif
if (c .eq. ‘i’ .and. lastc .eq. ‘e’) then
  totaleis = totaleis + 1
else if (c .eq. ‘e’ .and. lastc .eq. ‘i’) then
  totalies = totalies + 1
else
  lastc = c
endif
print *, ‘c = ‘, c, ‘lastc = ‘, lastc, ‘totalIEs = ‘, totalies,
  ‘totaleis = ‘, totaleis
goto 200
*
1000 print *, ‘the number of IEs is ‘, totalies
print *, ‘and the number of EIs is ‘, totaleis
end

#17 THE EXAMPLE WITH ASSERTION ADDED

else
  lastc = c
******************************************************************************
  if (lastc .ne. c) then
    print *, ‘lastc not equal to c’
    stop
  endif
******************************************************************************
#18 THE EXAMPLE ASSERTION AND TRACE

```c
... else
    lastc = c
C
***** atrace statement  ***************
    print *, 'lastc = ', lastc, 'c = ', c
C
******************************************************************************
C if (lastc .ne. c) then
C    print *, 'lastc not equal to c'
C    stop
C endif
C
******************************************************************************
```

#19 THE EXAMPLE DEBUGGED

```c
character*1 c, lastc
character*80 line
integer index
integer totalies, totaleis
C
totales = 0
totaleis = 0
lastc = char(0)
C
print *, 'Type any number of lines followed by EOF'
C
100 read (unit = *, fmt = 101, end = 1000) line
101 format (A80)
   index = 1
200 c = line(index:index)
   index = index + 1
   if (index .gt. 80) goto 100
   if (c .ge. 'A' .and. c .le. 'Z') then
      c = char(ichar(c) + 32)
   endif
   if (c .eq. 'i' .and. lastc .eq. 'e') then
      totaleis = totaleis + 1
   else if (c .eq. 'e' .and. lastc .eq. 'i') then
      totalies = totalies + 1
   endif
C else **** not necessary ****
lastc = c
C endif **** moved up ****
goto 200
C
1000 print *, 'the number of IEs is ', totalies
print *, 'and the number of EIs is ', totaleis
end
```