Program 3 Solution

CS-185, Spring '99, Dr. C. S. Tritt

Outline

```
Declare constants and variables.
Open input and output files and check for errors.
Loop (do) until every line in input file has been read.
    Read Width, Height, Frame, Style and Glass and test for errors and end-of-file.
    Calculate area and perimeter and start output.
    Process (select case) frame code, calculate frame cost and output frame information.
    Process (select case) style code, store style factor and output style information.
    Process (select case) glass code, calculate glass cost and output glass information.
    Calculate and output window cost.
    Add window cost to total, increment line number.
Write results, close files and stop.
```

Flowchart (done in Visio)

Click here to go to linked Acrobat (.PDF) file.

Pseudo Code

```
! Declare constants and variables...
! I/O Unit numbers and I/O status variable.

integer, parameter InUnit <- 10
integer, parameter OutUnit <- 11
integer IOStatus

! Parameters for frame, glass and style codes tests.

integer, parameter Wood <- 1, Vinyl <- 2, Aluminum <- 3
integer, parameter Standard <- 1, LowE <- 2
integer, parameter Budget <- 1, Regular <- 2, Premium <- 3

! Cost parameters.

real, parameter :: WdCost <- 2.25, VinylCost <- 1.75, AlumCost <- 2.75 ! $/ft
real, parameter :: StdGlassCost <- 2.50, LowEGlassCost <- 4.25 ! $/ft^2
real, parameter :: BdgtFactor <- 1.8, RegFactor <- 2.0, PremFactor <- 2.6

! Input variables.
```

```
real Width    ! Window width in feet.
real Height   ! Window height in feet.

integer Frame ! Frame code.
integer Glass ! Glass code.
integer Style ! Style code.

! Other variables.

real Perimeter ! Window perimeter (ft).
real Area ! Window area (ft^2).
real Factor ! Specified style cost factor for a particular window.
real FrameCost ! Calculated frame cost ($).
real GlassCost ! Calculated glass cost ($).
real Cost ! Calculated cost of a particular window ($).
real JobCost <- 0.00 ! Calcualted total cost of windows in a job.
integer Line <- 1 ! Current line being read.

! Open input and output files.

    open(unit = InUnit, file = 'job.dat', ...)
    if IOStatus > 0 then
        write(*,*) "Can't open input file.  Aborting."
        stop
    open(unit = OutUnit, file = 'job.out', ...)
        if IOStatus > 0 then
        write(*,*) "Can't open output file.  Aborting."
        stop

! Loop until every line in input file has been read.

    do

! Read data and test for errors and end-of-file.

        read(InUnit, *, ...) Width, Height, Frame, Style, Glass
        if IOStatus < 0 then
            exit
        else if IOStatus > 0 then
            write(*,*) 'Error reading line ', Line, '.  Aborting.'
            write(OutUnit,*) 'Error reading line ', Line, '.  Aborting.'
            close(OutUnit)
            close(InUnit)
```

```
      stop

! Calculate area and perimeter and start output.

    Area <- Width * Height
    Perimeter <- 2.0 * (Width + Height)
    write(OutUnit, ...) 'A ', Width, ' ft by ', Height, ' ft '

! Process frame type.

    select on Frame
      case Wood
        FrameCost <- Perimeter * WdCost
        write(OutUnit, ...) 'wood framed '
      case Vinyl
        FrameCost <- Perimeter * VinylCost
        write(OutUnit, ...) 'vinyl framed '
      case Aluminum
        FrameCost <- Perimeter * AlumCost
        write(OutUnit, ...) 'aluminum framed '
      case default
        write(*,*) 'Invalid frame code on line ', Line, '.  Skipping window.'
        write(OutUnit, *) 'Invalid frame code on line ', Line,'.  Skipping window.'
        cycle

! Process style information.

    select on Style
      case Budget
        Factor <- BdgtFactor
        write(OutUnit, ...) 'budget window'
      case Regular
        Factor <- RegFactor
        write(OutUnit, ...) 'standard window'
      case Premium
        Factor <- PremFactor
        write(OutUnit, ...) 'premium window'
      case default
        write(*, *) 'Invalid style code on line ', Line, '.  Skipping window.'
        write(OutUnit, *) 'Invalid style code on line ', Line,'.  Skipping window.'
        cycle

! Process glass type information.
```

```
    select on Glass
      case Standard
        GlassCost <- Area * StdGlassCost
        write(OutUnit, ...) 'with regular glass '
      case LowE
        GlassCost <- Area * LowEGlassCost
        write(OutUnit, ...) 'with Low-E glass '
      case default
        write(*, *) 'Invalid style code on line ', Line, '.  Skipping window.'
        write(OutUnit, *) 'Invalid style code on line ', Line, '.  Skipping window.'
        cycle

! Calculate and write window cost.

    Cost <- Factor * (FrameCost + GlassCost)
    write(OutUnit, ...) 'will cost $', Cost, '.'

! Add window cost to total, increment line number.

    JobCost <- Cost + JobCost
    Line <- Line + 1

! Write results, close files and stop.

  write(OutUnit, ...) 'The total cost of this order would be $', JobCost
  close(OutUnit)
  close(InUnit)
  stop
```

Source Code

```
program Windows
!
! The program calculates the cost of a set of custom
! windows.
!
! I/O Unit numbers and I/O status variable.

integer, parameter :: InUnit = 10
integer, parameter :: OutUnit = 11
integer :: IOStatus

! Parameters for frame, glass and style codes tests.

integer, parameter :: Wood = 1, Vinyl = 2, Aluminum = 3
integer, parameter :: Standard = 1, LowE = 2
integer, parameter :: Budget = 1, Regular = 2, Premium = 3
```

```
! Cost parameters.

real, parameter :: WdCost = 2.25, VinylCost = 1.75, AlumCost = 2.75 ! $/ft
real, parameter :: StdGlassCost = 2.50, LowEGlassCost = 4.25 ! $/ft^2
real, parameter :: BdgtFactor = 1.8, RegFactor = 2.0, PremFactor = 2.6

! Input variables.

real :: Width    ! Window width in feet.
real :: Height   ! Window height in feet.

integer :: Frame ! Frame code.
integer :: Glass ! Glass code.
integer :: Style ! Style code.

! Other variables.

real :: Perimeter ! Window perimeter (ft).
real :: Area ! Window area (ft^2).
real :: Factor ! Specified style cost factor for a particular window.
real :: FrameCost ! Calculated frame cost ($).
real :: GlassCost ! Calculated glass cost ($).
real :: Cost ! Calculated cost of a particular window ($).
real :: JobCost = 0.00 ! Calcualted total cost of windows in a job.
integer :: Line = 1 ! Current line being read.

! Open input and output files.

  open(unit = InUnit, file = 'job.dat', status = 'old', iostat = IOStatus)
  if (IOStatus > 0) then
    write(*,*) "Can't open input file.  Aborting."
    stop
  end if
  open(unit = OutUnit, file = 'job.out', status = 'unknown', iostat =
IOStatus)
  if (IOStatus > 0) then
    write(*,*) "Can't open output file.  Aborting."
    stop
  end if

! Loop until every line in input file has been read.

  do

! Read data and test for errors and end-of-file.

    read(InUnit,*, iostat = IOStatus) Width, Height, Frame, Style, Glass
    if (IOStatus < 0) then
      exit
    else if (IOStatus > 0) then
      write(*,*) 'Error reading line ', Line, '.  Aborting.'
      write(OutUnit,*) 'Error reading line ', Line, '.  Aborting.'
      close(OutUnit)
      close(InUnit)
      stop
    end if
```

```
! Calculate area and perimeter and start output.

    Area = Width * Height
    Perimeter = 2.0 * (Width + Height)
    write(OutUnit, '(a, 2(f6.2,a))', advance = 'no') 'A ', Width, ' ft by ', &
      Height, ' ft '

! Process frame type.

    select case (Frame)
      case (Wood)
        FrameCost = Perimeter * WdCost
        write(OutUnit, '(a)', advance = 'no') 'wood framed '
      case (Vinyl)
        FrameCost = Perimeter * VinylCost
        write(OutUnit, '(a)', advance = 'no') 'vinyl framed '
      case (Aluminum)
        FrameCost = Perimeter * AlumCost
        write(OutUnit, '(a)', advance = 'no') 'aluminum framed '
      case  default
        write(*,*) 'Invalid frame code on line ', Line, '.  Skipping window.'
        write(OutUnit, *) 'Invalid frame code on line ', Line, &
          '.  Skipping window.'
        cycle
    end select

! Process style information.

    select case (Style)
      case (Budget)
        Factor = BdgtFactor
        write(OutUnit, '(a)') 'budget window'
      case (Regular)
        Factor = RegFactor
        write(OutUnit, '(a)') 'standard window'
      case (Premium)
        Factor = PremFactor
        write(OutUnit, '(a)') 'premium window'
      case default
        write(*, *) 'Invalid style code on line ', Line, '.  Skipping window.'
        write(OutUnit, *) 'Invalid style code on line ', Line, &
          '.  Skipping window.'
        cycle
    end select

! Process glass type information.

    select case (Glass)
      case (Standard)
        GlassCost = Area * StdGlassCost
        write(OutUnit, '(a)', advance = 'no') 'with regular glass '
      case (LowE)
        GlassCost = Area * LowEGlassCost
        write(OutUnit, '(a)', advance = 'no') 'with Low-E glass '
      case default
        write(*, *) 'Invalid style code on line ', Line, '.  Skipping window.'
        write(OutUnit, *) 'Invalid style code on line ', Line, &
```

```
          '.  Skipping window.'
        cycle
    end select

! Calculate and write window cost.

    Cost = Factor * (FrameCost + GlassCost)
    write(OutUnit, '(a, f6.2, a)') 'will cost $', Cost, '.'

! Add window cost to total, increment line number and read another line.

    JobCost = Cost + JobCost
    Line = Line + 1
  end do

! Write results, close files and stop.

  write(OutUnit, '(a, f7.2)') 'The total cost of this order would be $',
JobCost
  close(OutUnit)
  close(InUnit)
  stop
end program Windows
```

Sample Input File

```
2.4 5.6 1 1 1
5.6 12.4 2 2 2
3.0 4.0 3 3 1
```

Sample Output File

```
A   2.40 ft by   5.60 ft wood framed budget window
with regular glass will cost $125.28.
A   5.60 ft by  12.40 ft vinyl framed standard window
with Low-E glass will cost $716.24.
A   3.00 ft by   4.00 ft aluminum framed premium window
with regular glass will cost $178.10.
The total cost of this order would be $1019.62
```