ProTeX Fortran Module Template

Here is the ASTG standard Fortran template showing the proper use of ProTeX tags in a module:

```
!-----------------------------------------------------------------------------
! NASA/GSFC, Software Integration & Visualization Office, Code
610.3
!-----------------------------------------------------------------------------
!BOP
! !MODULE: MyModule_mod
!
! !AUTHOR:
! TODO_name, TODO_affiliation
!
! !DESCRIPTION:
!
TODO_describe_purpose_of_module,_ensuring_there_are_TWO_blank_lines_before_
! the_next_PROTEX_keyword.
!
!
! !REVISION HISTORY:
! TODO_dd_mmm_yyyy - TODO_describe_appropriate_changes - TODO_name
!-----------------------------------------------------------------------------
! !INTERFACE:
module MyModule_mod

   ! !USES:
   use AnotherModule_mod

   implicit none


  private


   ! !PUBLIC TYPES:


  public
 :: SomeType              ! TODO_brief_description

   ! !PUBLIC MEMBER FUNCTIONS:


  public
 :: someFunction          ! TODO_brief_description

   ! !PUBLIC DATA MEMBERS:
   ! NOTE_avoid_if_possible

!EOP
!-----------------------------------------------------------------------------
```

```
contains

!-----------------------------------------------------------------------------
!BOP
! !IROUTINE: TODO_function_routine_name
!
! !AUTHOR:
! TODO_name_and_affiliation
!
! !DESCRIPTION:
! TODO_describe_routine_purpose_followed_by_TWO_blank_lines.
!
!
! !REVISION HISTORY:
! DD_MMM_YYYY TODO_describe_appropriate_changes.
!---------------------------------------------------------------------------

! !INTERFACE:
   function someFunction(inParam, outParam, inOutParam)

      ! !USES:
      use AnotherModule_mod



  private


      ! !INPUT PARAMETERS:
      real, intent(in) :: inParam         ! TODO_description

      ! !OUTPUT PARAMETERS:
      real, intent(out) :: outParam       ! TODO_description

      ! !INPUT/OUTPUT PARAMETERS:
      real, intent(inout) :: inOutParam   ! TODO_description

      ! !RETURN VALUE:                    !(Used in functions)
      real :: someFunction                ! TODO_description

      ! !LOCAL VARIABLES:
      real :: someVariable                ! TODO_description
      !EOP

      !BOC
      ! TODO_insert_code_here

      !EOC
   end function someFunction

end module MyModule_mod
```

And here's a simple example of it's use:

```
!-----------------------------------------------------------------------

! NASA/GSFC, Software Integration & Visualization Office, Code
610.3
!-----------------------------------------------------------------------
!BOP
! !MODULE: TriangleOperations
!
! !AUTHOR:
! I. M. Coder, A. Programmer - Superior Software, Inc.
!
! !DESCRIPTION:
! Calculate the attributes of a triangle.
!
!
! !REVISION HISTORY:
! 11 Apr 2007 - Initial version
!-----------------------------------------------------------------------
! !INTERFACE:
module TriangleOperations_mod

   ! !USES:
   use RectangleOperations_mod

   implicit none


  private


   ! !PUBLIC MEMBER FUNCTIONS:


  public
 area

!EOP
!-------------------------------------------------------------------------------

contains

!---------------------------------------------------------------------------

!BOP
! !IROUTINE: area
! Note that
"IROUTINE" is used

  for
 internal routines
!
! !AUTHOR:
! A. Programmer
```

```
!
! !DESCRIPTION:
! Calculate the area inside a triangle
!
!
! !REVISION HISTORY:
! None
!-----------------------------------------------------------------------------
! !INTERFACE:
   function area (sideX, sideY, sideZ)

      ! !USES:
      use config_mod        ! System wide configuration



  private


      ! !INPUT PARAMETERS:
      real, intent(in) ::  sideX,  & ! length of a side in
meters
                           & sideY,  & ! length of a side in
meters
                           & sideZ     ! length of a side in
meters

      ! !RETURN VALUE:
      real :: area                  ! function returns interior
area

      ! !LOCAL VARIABLES:
      real :: theta                 ! Angle between sideX &
sideY
      real :: height                ! Height perpendicular to sideY
      !EOP

      !BOC
      theta = acos(                                  &
         &(sideX ** 2 + sideY ** 2 - sideZ ** 2) /  &
         &(2.0 * sideX * sideY) )
      height = sideX * sin(theta)
      area = 0.5 * sideY * height
      !EOC

   end function area

end module TriangleOperations_mod
```