Bugs and Debug - Dealing with Programming Mistakes

1. Bugs - Program Errors
   - Syntax error - Bugs that can be caught by Compilers

Can you fix the syntax errors in the above source code with the help of the compilation error messages?
Run-time error - Bugs that can be caught by Operating system in run-time

Can you interpret the error message from the OS? Can you further tell which statement cause the problem?

Logic error - Bugs that can only be caught by Users in run-time

Does the program make sense to you? Can you fix the problem in the source code?
2. Debug - The process of correcting program errors
   - Make the best use of Compilers:
     - IMPLICIT NONE statement + Data Declaration

Can you find the typo that causes the logic error? If you cannot catch it in 10 seconds, try to insert `IMPLICIT none` Statement in between the PROGRAM Statement and the Data Declaration Statement, then recompile it. Observe the error message from the compiler. [hypotenuse.f95 buggy version]

- Simplex algorithm design + clean coding style
  - **Simple algorithm design** is the key to logical bug free
  - **Variable Initialization**: Life-saver Coding Habit
    
    ```fortran
    integer:: a=0
    ```

- **Meaningful Names** for identifiers
- **Consistent Indentations** for blocked statements
  
  Use Emacs Auto-indentation: click "F90->Indent Subprogram" from the menu

- **Necessary Comments** to keep you fresh
  
  `!` anywhere in a line starting with the exclamation mark "!
  `!` will be ignored by the compiler.
  `!` so you can put extra comments inside your source code to make it easier to read

- Manual logical bug predator: Inserting Write statement

  ```fortran
  Program main
  real:: sideA=0., sideB=0., hypotenuse
  write(*,*) 'lecture 10 Intrinsic function demo -J.Xue'
  write(*,*) 'Please enter side A and side B of a right triangle'
  read(*,*) sideA, sideB
  hypotenuse = sqrt(sideA**2+sideB**2)
  write(*,*) 'sideA=', sideA
  write(*,*) 'sideB=', sideB
  write(*,*) 'hypotenuse = ', hypotenuse
  ```
Write(*,*) "sideA = ", sideA
Write(*,*) "sideB = ", sideB
Write(*,*) "hypotenuse = ", hypotenuse
End Program

- Ultimate logical bug predator: Debugger program -- GDB
  - Add `-g` option to the compiling command first. For example:
    ```
g95 hypotenuse.f95 -o hypotenuse.exe -g
  ```
  - In Emacs, activate the source file editing window, then activate gdb debugger by the command (Use the default configurations and arguments.)
    ```
    M-x gdb
    ```
  - Click menu **Gud->GDB-UI->Display Other Windows**, then you'll see similar layout to the example below.

An example of debugging a program inside **GDB** debugger.
- Upper left window shows **gdb command line control**;
- Upper right window shows current variables in memory;
- Middle window shows source code with break point (red dot), and current progress indicator (black triangle) on the left side of the window;
- Bottom left window shows the stack of subprogram calls at the current moment;
- Bottomright window shows breakpoint list.

Last update:2009-09-15