Guidelines

To complete assignments you must hand in a report including:

1. Title
   - The name of the homework assignment. Ex. “Bisection Method”

2. Description of Problem
   - Description of the problem with a discussion of relevant mathematics. Ex. Explain how and why the bisection method works. Be specific and include a few steps worked out by hand with explanations of each step.

3. Description of the Program
   - Description of your program which describes the algorithm you used and details your implementation. Ex. Explain how you implement the bisection method. You may also copy the relevant piece of code to help explain your implementation.

4. Results and Conclusions
   - Discussion of the results including any tables or figures needed. Ex. Tell me all the roots for the two equations with error bounds. You must also explain why you believe the answer is correct. The correct answer alone is not enough, you have to convince me that it is correct to get credit for it.

5. Program Listing
   - Include all your source code, makefiles and instructions on how to execute your code.

You must also email me your source code, makefiles and instructions on how to execute your code.

- Email me at pgarreau@math.fsu.edu
- Include [MAT5939-03] in the subject line of the email.
5 Timer Module

1. Write a program that creates a two dimensional array with dimensions 2000 by 2000.
2. Assign random values to the array (Sample Code available below).
3. Write a module to time your code.
   - The module must use the Date_and_Time Fortran Intrinsic.
   - The module must include a derived type for the timer.
   - The module must include functions or subroutines to start the timer, stop the timer and return the elapsed time.
4. Use this module to time how long it takes to multiple the 2000 by 2000 array against itself 5 times.

Algorithm 2 Pseudo code: timer module
1: array ← 2000 × 2000 r.n.
2: Start timer
3: for i = 1, . . . , 5 do
4:   array = array × array
5: end for
6: Start timer
7: Output: elapsed time

Example Using Random Numbers
PROGRAM RandomEx
  use double
  IMPLICIT NONE
  Real(kind=dp) :: array(2000,2000)
  CALL init_random_seed()
  CALL RANDOM_NUMBER(array)
END PROGRAM RandomEx

SUBROUTINE init_random_seed()
  IMPLICIT NONE
  INTEGER :: i, n, clock
  INTEGER, DIMENSION(:), ALLOCATABLE :: seed
  CALL RANDOM_SEED(size = n)
  ALLOCATE(seed(n))
  CALL SYSTEM_CLOCK(COUNT=clock)
  seed = clock + 37 * (/ (i - 1, i = 1, n) /)
  CALL RANDOM_SEED(PUT = seed)
  DEALLOCATE(seed)
END SUBROUTINE