Task 1 Write-up

I. Code explanation

MonoVariable:

There are two private class members.
The “storage” of type “T” which means it is a generic type and it take the responsibility of storing data for the data structure(MonoVariable).
The other one is “occupied” of boolean type, it is an indicator indicating the data structure is occupied or not.

There are two public functions.
The function “void becomes(T val)” does a validation checking right after the function call has made and print some error message then terminate the function by return if the argument “val” is null. After the validation checking, comes a loop of “wait()” to block thread when the mono variable is occupied. When the data structure is not occupied, it will store the data by assigning the “val”’s reference to “storage” and set “occupied” to true. Call “notify()” at last.
The function “T consume()” is designed to return the “storage”’s reference to thread and set the mono variable to not occupied and if the mono variable is not occupied, it will keep blocking using “wait()” until the “occupied” is true. Deleting the “storage” is not necessary as “occupied” has been set to true and next operation must override the “storage”. Call “notify()” at last.

II. Test explanation

A. Single Thread Test
   1. Basic Functionality
      (1) becomes() then consume()
      (2) becomes()
      (3) consume()
   2. Null Test
      (1) Customised error message presented, abort the becomes().

B. Multi-Thread Test
   1. Basic Functionality
      (1) Two different arithmetic threads performing “add one” and “time two”.
      (2) Preset the MonoVariable value to be 1
      (3) In each thread, consume the storage in the MonoVariable first, then do the calculation, becomes() at last.