Instructions: You need to install and run Dafny for problem 4. Instructions for installing Dafny are available here: https://dafny.org/latest/Installation
Turn in a hard copy of your answers. Additionally, for problem 4, send an ASCII text file via email to bultan@ucsb.edu (with subject line “272 HW2”) containing your Dafny code and the outputs generated by Dafny.

1. Consider a class called BufferControl that keeps track of the number of items in a buffer, the number of items that have been inserted to the buffer, and the number of items that have been removed from the buffer. To simplify things we do not keep track of the items in the buffer in this class. BufferControl class has three integer fields: numItems, numInserted, and numRemoved. The constructor BufferControl() sets all three fields equal to zero. BufferControl class has three methods: void insert(), void remove(), and int getNumItems(). getNumItems() returns the value of numItems and does not change the values of any fields. When insert() method is called numItems and numInserted are incremented by one and numRemoved remains the same. When delete() method can only be can only be called when numItems is greater than or equal to 1. When delete() method is called numItems is decremented by one and numRemoved is incremented by one and numInserted remains the same.

(a) Write the contract for the BufferControl class in JML by writing the pre and post-conditions for each method. Also write the (strongest) class invariant.

(b) Assume that there is another class called BoundedBufferControl. BoundedBufferControl class is very much like the BufferControl except that it has another integer field called size. The constructor BoundedBufferControl(int s) requires the input value s to be greater than or equal to 1, sets size to the input value s and initializes the rest of the variables to zero. The value of the variable size is not modified by any methods after construction. The behaviors of the methods remove() and getNumItems() for BoundedBufferControl are identical to those of corresponding methods in BufferControl. The method insert() can only be called when numItems is strictly less than size, otherwise it behaves exactly like the insert() method of the BufferControl.

Write the contract for the BoundedBufferControl class in JML by writing the the pre and post-conditions for each method. Also write the (strongest) class invariant.

(c) Consider the following two class structures: 1) BufferControl is superclass of BoundedBufferControl. 2) BoundedBufferControl is superclass of BufferControl. Based on the inheritance rules in design by contract, explain whether these options would or would not work and why.
2. Prove the following Hoare triples. Show the axioms and the inference rules you use.

(a) \( \{ x > 5 \lor x < -2 \} \text{ if } (x>0) \text{ then } y:=x \text{ else } y:=-x \{ y \geq 0 \} \)

(b) \( \{ \text{true} \} \text{ result:=1; i:=1; while } (i\leq10) \text{ (result:=result*i; i:=i+1) } \{ \text{result} = 10! \} \)

3. Compute the following weakest preconditions (show each step of the derivation).

(a) \( \text{wp}(i:=i*2; \ j:=j-3, i + j = 0) \)

(b) \( \text{wp}(\text{if } (x>y) \text{ then } z:=x \text{ else } z:=y, z \geq x) \)

4. Install and use the Dafny tool for the following:

(a) Write a Dafny method that takes an integer array and returns the index of the first negative element in the array using a loop that traverses the array (if there is no negative element return -1). Prove the correctness of the method you wrote using Dafny by writing pre and post-conditions and loop invariants for the method you wrote.

(b) Write a Dafny method that takes an integer \( x \) and an integer \( n \) as input, computes \( x \) raised to the power \( n \) using a loop, and returns the result. Prove the correctness of the method you wrote using Dafny by defining a power function recursively, and writing pre and post-conditions and loop invariants for the method you wrote.