1. For each of these parameter passing mechanisms,
   (a) pass by value
   (b) pass by reference
   (c) pass by value-result
   (d) pass by name

   state what the following program (in some Pascal-like language) would print if that parameter
   passing mechanism was used:

   ```pascal
   program foo;
   var i,j: integer;
   a: array[1..5] of integer;

   procedure f(x,y:integer)
   begin
   x := x * 2;
   i := i + 1;
   y := a[i] + 1;
   end

   begin
   for j := 1 to 5 do a[j] = j;
   i := 1;
   f(i,a[i]);
   for j := 1 to 5 do print(a[j]);
   end.
   ```

2. (a) Define the terms static scoping and dynamic scoping.
   (b) Give a simple example, in any language you like (actual or imaginary), that would
   illustrate the difference between static and dynamic scoping. That is, write a short piece
   of code whose result would be different depending on whether static or dynamic scoping
   was used.
   (c) In a block structured, statically scoped language, what is the rule for resolving variable
   references (i.e. given the use of a variable, how does one find the declaration of that
   variable)?
   (d) In a block structured but dynamically scoped language, what would the rule for resolving
   variable references be? (This wasn’t covered in class, but it follows from the definition
   of dynamic scoping.)

3. (a) Define the following terms:
   i. Abstract Data Type
ii. Inheritance

iii. Subtyping

iv. Dynamic Dispatch

(b) In Ada, write some code to define an abstract data type. Just include the necessary declarations, don’t include any executable code.

4. (a) In Ada, write a procedure that declares two tasks, task one and task two. When the procedure is called, task one and task two should print “one” and “two”, respectively, over and over such that the printing of “one” and “two” is perfectly interleaved. Thus, the output should look like:

one
two
one
two
...

(b) Looking at the code you wrote for part (a), are the printing of “one” and the printing of “two” occurring concurrently? Justify your answer by describing what concurrency is and why these two events do or do not occur concurrently.

5. (a) In Scheme, write the function member that takes a value and a list and returns true if the value is in the list. For example,

(member 'x '(a b x y 1 2))
returns #t (true) and

(member 'x '(a b y 1 2))
returns #f (false). member should not search for the value within nested lists. (Note: This code should be about 4 lines)

(b) In ML this time (not Scheme), assume there is already a function called member that behaves like the one above (but only works on homogeneous lists, of course). Write the code for the function union that takes two lists, L1 and L2, and returns a list containing all the elements of L1 and L2 except that no values are duplicated. For example,

union [1,2,3,4] [3,4,5,6]

might return [1,2,3,4,5,6]. The order of the elements in the resulting list is not important, so your union can return the elements in any order you want. (Note: This code should be about 4 lines or so)

(c) What is the type of union, above?

6. In C++, multiple inheritance arises because it is quite natural to express a class as being related to two other classes. For example, a sea-plane could be considered both an airplane and a boat, and thus it’s natural to derive class seaplane from class airplane and class boat.

Java doesn’t have multiple inheritance. Describe what construct(s) in Java can be used to express multiple relationships, as seen in the sea-plane example (“a sea-plane is both a boat and an airplane”). Give an example in Java of how this construct(s) would be used instead of multiple inheritance.