ECE 322

SOFTWARE TESTING AND MAINTENANCE

Fall 2021

Assignment #5

Due date: Monday, November 22, 2021 by 11:00 PM

Total: 40 points

10 points

1.For the code shown below, draw a control flow graph and determine its cyclomatic complexity. Consider two situations:
(a)when compound decisions (such as those shown in lines 4, 9, and 11) are treated *en bloc*
(b)when the individual conditions in the compound decisions are treated separately.

```
1 output ("Enter 3 integers")
2 input (a, b, c)
3 output("Side a,b c: ", a, b, c)
4 if (a < b) and (b < a+c) and (c < a+b)
5 then isTriangle ← true
6 else isTriangle ← false
7 fi
8 if isTriangle
9 then if (a = b) and (b = c)
10 then output ("equilateral")
11 else if (a ≠ b ) and ( a ≠ c ) and ( b ≠ c)
12 then output ("scalene")
13 else output("isosceles")
14 fi
15 fi
16 else output ("not a triangle")
17 fi
```

10 points

2.For the piece of code shown below show *def-clear* paths for the variables *tv*, *av*, and *sum*.

```
public static double ReturnAverage(int value[],
                        int AS, int MIN, int MAX){
  /*
   Function: ReturnAverage  Computes the  average
   of all  those  numbers in the  input array  in
   the  positive  range  [MIN, MAX]. The  maximum
   size  of the array is AS. But, the  array size
   could be smaller than AS in which case the end
   of input is represented by -999.
  */
     int i, ti, tv, sum;
     double av;
     i = 0; ti = 0; tv = 0; sum = 0;
     while (ti < AS && value[i] != -999) {
         ti++;
         if (value[i] >= MIN && value[i] <= MAX) {
            tv++;
            sum = sum + value[i];
         }
         i++;
     }
     if (tv > 0)
        av = (double)sum/tv;
     else
        av = (double) -999;
     return (av);
}
```

10 points

3. Write a function in Python such that the *all-uses* coverage criterion produces more test cases than the *branch* coverage criterion.

10 points

4.Using the modified condition/branch coverage criterion, propose test cases for the following expression

$$(a \,||\, b) \;\&\&\; (\text{not}(c) \,||\, \text{not}(d))$$

Note that the test set is not unique; show all possibilities.