Submission Guidelines:

Due: 11:59pm ending Wednesday, July 4, 2018.

- The assignment should be submitted via Blackboard.
- The answers must be typed as a document.
- Make sure your name and your student ID are listed in your document.
- Name files as assignment4_<net-id>.<format>
- Accepted document formats are (.pdf, .doc or .docx). If you are using OpenOffice or LibreOffice, make sure to save as .pdf or .doc
- Please do not submit .txt files.
- If there are multiple files in your submission, zip them together as assignment4_<net-id>.zip and submit the .zip file.
- The maximum points one can get in this assignment is 100.
- You may resubmit the submit at any time. Late submissions will be accepted at a penalty of 10 points per day. Maximum latency is 3 days beyond which a grade of zero will be assigned. This penalty will apply regardless of whether you have other excuses.

Assignment Specifications:

1. (**13 pts.**) Draw a Structural Graph that represents the below piece of source code. You must indicate how each node and edge corresponds to the code. Then, find the test paths (*First you need to find all the prime paths and then you can create the test paths from them*)

```
int count_spaces(char* str)
{
  int length , i , count;
   count = 0;
   length = strlen(str );
   for(i=1; i<length; i++) {
     if(str[i] == ' ') {
        count++;
     }
   }
}
```

2. (**24 pts.**) Consider the following fragment of code:

```
public int foobar(int x, int y,int z) {
  int k = 0;
  for(int i=0; i < z; i++)
  {
     for(int j=0; j<y; j++)
      {
        if(i<j) {
          k=k*y+x;
        }
        else {
        k--;
        }
     }
   }
  return(k);
}
```

a) Draw a structural control flow graph that represents the above piece of source code. (5 pts.)

b) For you graph, enumerate a complete set of paths that guarantee node coverage. (5 pts.)

c) Construct a finite set of test cases that guarantee node coverage. For each test case indicate which execution path is covered. (5 pts.)

d) For the above fragment of code is there any difference in requiring complete edge coverage rather than node coverage? Justify your answer. (5 pts.)

e) Enumerate all the prime paths of the graph you have constructed. (5 pts.)

f) For each prime path that you have found construct a test case for that path. Indicate which prime path is covered by your test cases. (5 pts.)

3. Below are four graphs, each of which is defined by the sets of nodes, initial nodes, final nodes, edges, and defs and uses. Each graph also contains a collection of test paths. Answer the following questions about **each graph (12 points for each graph – Total 48 pts.)**

a) Draw the graph.

b) List all of the du-paths with respect to x. (Note: Include all du-paths, even those that are sub-paths of some other du-path).

c) For each test path, determine which du-paths that test path tours. For this part of the exercise, you should consider both direct touring and side-trips.
Hint: A table is a convenient format for describing this relationship.

d) List a minimal test set that satisfies all-defs coverage with respect to x. (Direct tours only.) Use the given test paths.

e) List a minimal test set that satisfies all-uses coverage with respect to x. (Direct tours only.) Use the given test paths.

f) List a minimal test set that satisfies all-du-paths coverage with respect to x. (Direct tours only.) Use the given test paths.

Graph I.
$N = \{0, 1, 2, 3, 4, 5, 6, 7\}$
$N_0 = \{0\}$
$N_f = \{7\}$
$E = \{(0,1), (1,2), (1,7), (2,3), (2,4), (3,2), (4,5), (4,6), (5,6), (6,1)\}$
$def(0) = def(3) = use(5) = use(7) = \{x\}$

Test Paths:
$t1 = [0, 1, 7]$
$t2 = [0, 1, 2, 4, 6, 1, 7]$
$t3 = [0, 1, 2, 4, 5, 6, 1, 7]$
$t4 = [0, 1, 2, 3, 2, 4, 6, 1, 7]$
$t5 = [0, 1, 2, 3, 2, 3, 2, 4, 5, 6, 1, 7]$
$t6 = [0, 1, 2, 3, 2, 4, 6, 1, 2, 4, 5, 6, 1, 7]$

Graph II.
$N = \{1, 2, 3, 4, 5, 6\}$
$N_0 = \{1\}$
$N_f = \{6\}$
$E = \{(1,2), (2,3), (2,6), (3,4), (3,5), (4,5), (5,2)\}$
$def(x) = \{1, 3\}$
$use(x) = \{3, 6\}$ // Assume the use of x in 3 precedes the def
Test Paths:
$t1 = [1, 2, 6]$
$t2 = [1, 2, 3, 4, 5, 2, 3, 5, 2, 6]$
$t3 = [1, 2, 3, 5, 2, 3, 4, 5, 2, 6]$
$t4 = [1, 2, 3, 5, 2, 6]$

Graph III.
$N = \{1, 2, 3, 4, 5, 6\}$
$N_0 = \{1\}$
$N_f = \{6\}$
$E = \{(1,2), (2,3), (3,4), (3,5), (4,5), (5,2), (2,6)\}$
$def(x) = \{1, 4\}$
$use(x) = \{3, 5, 6\}$

Test Paths:
$t_1 = [1, 2, 3, 5, 2, 6]$
$t_2 = [1, 2, 3, 4, 5, 2, 6]$

Graph IV.
$N = \{1, 2, 3, 4, 5, 6\}$
$N_0 = \{1\}$
$N_f = \{6\}$
$E = \{(1,2), (2,3), (2,6), (3,4), (3,5), (4,5), (5,2)\}$
$def(x) = \{1, 5\}$
$use(x) = \{5, 6\}$ // Assume the use of x in 5 precedes the def
Test Paths:
$t1 = [1, 2, 6]$
$t2 = [1, 2, 3, 4, 5, 2, 3, 5, 2, 6]$
$t3 = [1, 2, 3, 5, 2, 3, 4, 5, 2, 6]$

4. (**15 pts.**) Use the following program fragment for questions a-e below.

```
w = x;          // node 1
if (m > 0)
{
   w++;         // node 2
}
else
{
   w=2*w;       // node 3
}
// node 4 (no executable statement)
if (y <= 10)
{
   x = 5*y;     // node 5
}
else
{
   x = 3*y+5;   // node 6
}
z = w + x;      // node 7
```

a) Draw a control flow graph for this program fragment. Use the node numbers given above.

b) Which nodes have defs for variable w?

c) Which nodes have uses for variable w?

d) Are there any du-paths with respect to variable w from node 1 to node 7? If not, explain why not. If any exist, show one.

e) Enumerate all of the du-paths for variables w and x.