Assignment 5: Software Testing

Question 1

You are provided the following method that implements the insertion-sort algorithm on an array of integers.

```
4     void insertionSort(int array[]) {
5         int size = array.length;
6         for (int step = 1; step < size; step++) {
7             int key = array[step];
8             int j = step - 1;
9             while (j >= 0 && key < array[j]) {
10                array[j + 1] = array[j];
11                --j;
12            }
13            array[j + 1] = key;
14        }
15    }
```

(a) Draw a flow graph for the above method as follows: first, give line numbers for the above code. Then, draw a flow graph and label each node with the line number(s) it corresponds to. Provide unique names (i.e. 'a', 'b', etc.) for each node separately so that they can be referred to. **(5 points)**

(b) Provide all basis paths in the above flow graph. Specify each basis path in terms of the unique number for each constituent node. **(20 points)**

The "insertionSort" method presented here is formally written in a Java file – named InsertionSort.java (attached to the assignment).:

Question 2

Convert each of the identified basis paths into a Junit test case. Create a Junit test class – a Java file, name InsertionSortTest.java. In this file, write the test cases (e.g., testingEmptyArray) for each basis path. Please give meaningful names to the test cases. On top of each test case, in Java comments, indicate which path is the test case testing – please see the sample template. **Use the InsertionSort.java file from Question 1 to ensure your test cases execute correctly. (25 points).**

```
public class InsertionSortTest {
```

```
  @Before
  public void setUp(){

  }

  /*
   * Basis Path Testing - Path 1 (Please see the answer to question 1)
   */
  @Test
  public void testingEmptyArray(){
    InsertionSort is = new InsertionSort();
  }


  @After
  public void cleanUp(){

  }

}
```

What to submit

Please submit the following on ReggieNet.

- A **PDF** document for Question 1
- InsertionSortTest.java for Question 2