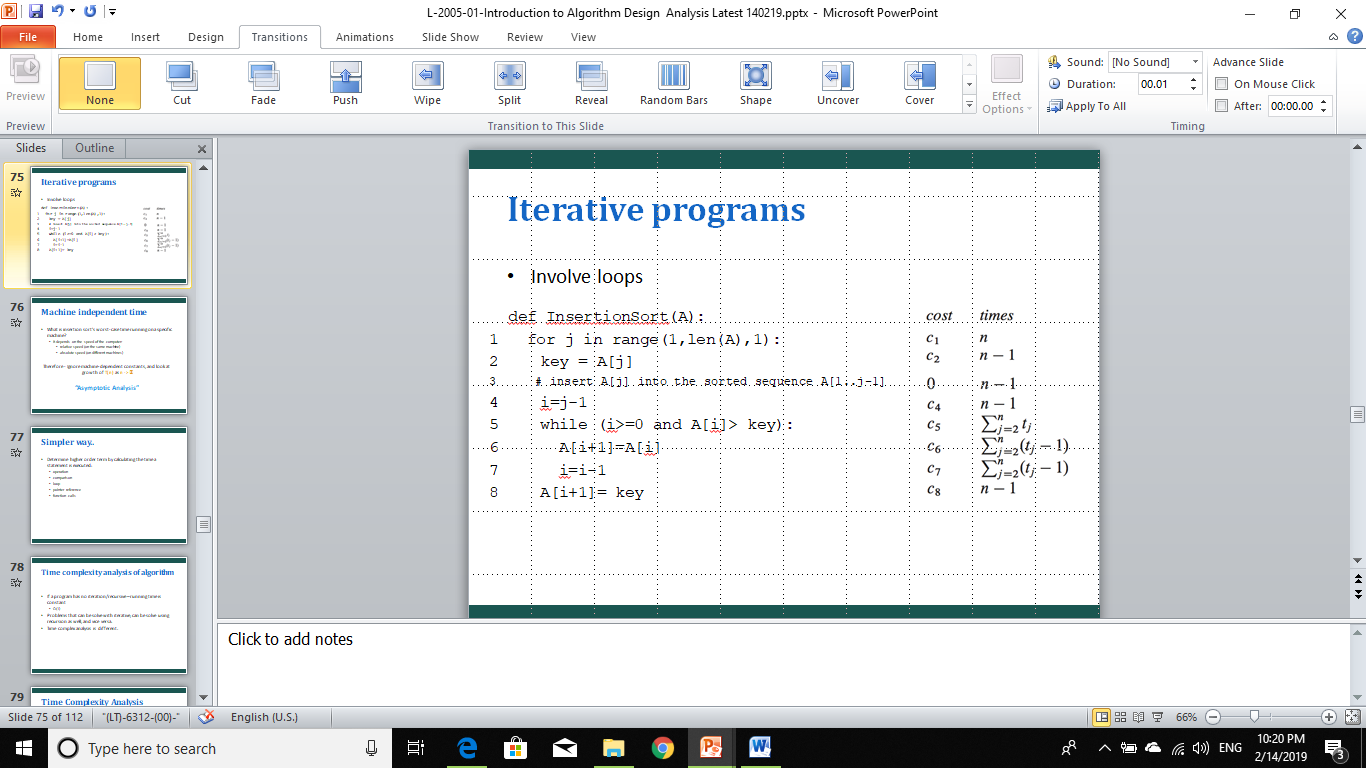
**WIA2005 Algorithm Design &Analysis**

**Semester 2**

**Tutorial 1**

1. The following is an insertion sort algorithm.



Illustrate the insertion sort operation on array A = 41, 51, 69, 36, 51, 68.

1. Modify the insertion sort algorithm to sort array into decreasing order.
2. Write a pseudocode for linear search for the following requirement:

**Input:** A sequence of *n* numbers A = 〈a1, a2, …, an〉 and a value *v*.

**Output:** An index *i* such that *v* = A[*i*] or the special value NIL if *v* does not appear in A.

1. Express the function n3 / 1000 – 100n2 – 100n + 3 in terms of Θ-notation.
2. For the following pairs of functions, f(n) and g(n), determine if they belong to Case 1: f(n) = O(g(n)) or Case 2: g(n) = O(f(n)). Formally justify your answer.
   1. f(n) = 3n +2 , g(n) = n
   2. f(n) = (n2 – n)/2 , g(n) = 6n
   3. f(n) = n+2√n , g(n) = n2
   4. f(n) = n2 + 3n + 4 , g(n) = n3
3. Given the iterative function below (in Java), calculate their time complexity.
   1. function1 (){

for (int i = 1; i <= n; i ++) {

printf(“Hello world”);

}

}

* 1. function2(){

for (int i = 1; i <=n; i ++) {

for (int j = 1; j <=n; j ++) {

printf(“Hello world”);

}

}

* 1. function3 (){

for (int i = 1; i2 <= n; i ++) {

printf(“Hello world”);

}

}

* 1. function4 (){

for (int i = 1; i <= n; i = i\*2) {

printf(“Hello world”);

}

}

* 1. function3(){

for (int i = n/2; i <=n; i ++) {

for (int j <= 1; j <=n/2; j = 2\*j) {

for (int k = 1; k <= n; k\*2) {

printf(“Hello world”);

}

}

}

}

}