CS1100 - Lecture 2

Instructor : Jasine Babu

Teaching Assistants : Nikhila K N, Veena Prabhakaran

1 Introduction

In the last class, we have studied about *while* statement, which is one of the simplest iterative constructs. It is used for repeatedly executing a set of instructions based on the result of evaluation of a condition.

In this lecture, we will continue on this topic, by taking another example. A program for finding maximum of n numbers is given below.

```
int n,a,max,counter
Input n
counter <-- 0
if( n==0 )
{
    output("No numbers to compare")
}
else
{
    Input a
    max <-- a
    counter <-- 1
    while( counter < n)
    {
      Input a
      if( a > max )
      {
         max <-- a
      }
      counter <--counter+1
    }/*end of while */

    output max

}/* end of else */

/* end of the program */
```

This program first reads the value of variable n. The program uses a variable `counter` to keep track of the number of values (other than `n`) seen so far. The variable `counter` is initialized to 0. When the control reaches the statement `if(n==0)`, the value of variable

`n` and the value 0 are taken to the ALU and compared. When the condition evaluates to *true*, then it means there are no numbers to compare. In this case, the program outputs "No numbers to compare" and ends. If the condition evaluates to `false`, then it will read a number and initializes the `max` value as that number and increment the counter to 1. When the control reaches the `while( counter<n )`, the current value of `counter` and `n` taken into ALU and compared. When the condition evaluates to true, it means we have more numnbers to input. Then the instructions between the opening bracket { after the while instruction and its pairing closing bracket } are to be executed until the condition becomes false. Inside the loop, a new number is taken as input in variable `a` and it is compared with the current value of `max`. If `a > max`, the value of `max` is updated as the value of `a`. Before ending the loop, the variable `counter` is incremented.

When the condition `(counter<n )` evaluates to false, it means all numbers are already entered. So, at this point, `max` has the maximum of all n numbers. After this, the value of `max` is given as output to the user.