**Lebanese American University**

# Department of Computer Science & Mathematics

**CSC 322 – Computer Organization Lab**

**Charbel Daoud**

****

**Final Project**

**Rebecca Tannous**

**201905551**

**Submitted on 22/05/2020**

**Project:** Program that translates basic Java instructions to assembly

The program starts by reading a text file. In addition to reading the first line, the program reads the next line to check if there is an else statement. After reading each line, the Main class calls a method type(String S) which returns an integer that represents a specific instruction. After checking the type of the instruction an appropriate method is called.

In the Class Assembly, the method ***RIType***(String S) prints the corresponding assembly code. This method covers I and R-type instructions. It checks for several cases like if it’s a regular Java statement, shorthand operation (+=…), increment/decrement…

ForLoop(String S) is another method of this class. First we split the string according to two delimiters which are ‘{’ and ‘;’. After initializing the integer that controls the loop to zero at each statement the method RIType is called.

While(String S) also splits the strings then translates the instructions line by line. It cover several cases : < , > , <= , >= , == and != with both integers and registers.

Conditions, this method splits the string then checks if there is an else statement. It covers several cases like <,> : < , > , <= , >= , == and != with both integers and registers. After writing the appropriate assembly code for the condition it translates the code line by line using RIType method

Method Type(String S) checks for specific words in the string to determine the type of the instruction.

Two methods getRegister are used to get the name of the register. The methods return a String containing “X” in addition to the index of the register. To obtain the index of the register a vector containing all the letter of the alphabet is used and we just use the method .indexOf to get the index. This method guarantees a logical use of registers even if registers appear several times in an instruction.

For the code to work properly a specific format of the file should be used:

a = a + b; (spaces between characters)

a += b;

c = a + 2

while ( x == 4)

**Test Cases:**

