QUIZ #05
CSCI-410 Spring 2013

1. (16pts) Write the VM code for each of the six variants of the test conditional shown. You may assume that all variables exist in the file "example.vm" and are static.

```
if (test)
   if_code     // Placeholder label - use instead of VM code
else
   else_code  // Placeholder label - use instead of VM code

assume a => static[0], b => static[1]
```

(test) is (a EQ b)	(test) is (a LT b)	(test) is (a GT b)
push static 0	push static 0	push static 0
push static 1	push static 1	push static 1
eq	lt	gt
if-goto IF_CODE	if-goto IF_CODE	if-goto IF_CODE
// else code here	// else code here	// else code here
goto IF_END	goto IF_END	goto IF_END
label IF_CODE	label IF_CODE	label IF_CODE
// if code here	// if code here	// if code here
label IF_END	label IF_END	label IF_END

(test) is (a NE b)	(test) is (a GE b)	(test) is (a LE b)
push static 0	push static 0	push static 0
push static 1	push static 1	push static 1
eq	lt	gt
if-goto ELSE_CODE	if-goto ELSE_CODE	if-goto ELSE_CODE
// if code here	// if code here	// if code here
goto IF_END	goto IF_END	goto IF_END
label ELSE_CODE	label ELSE_CODE	label ELSE_CODE
// else code here	// else code here	// else code here
label IF_END	label IF_END	label IF_END

Note that there are a few reasonable alternatives. The approach here is to simply swap the locations of the "if" code and the "else" code in order to negate the relation. Another alternative is to use the "neg" command following the relational command. By using the "neg" command on the top row (and not the bottom), the "if" code can always be kept first, making the code generation for bare "if" statements (i.e., no "else" block) easy. For the inequalities, the "neg" command can be avoided by swapping the order in which operands are pushed onto the stack. Any of these are reasonable approaches.

2. (4pts) Write the VM code for the following loop. You may assume that all variables exist in the file "example.vm" and are static.

```
a = -20
b = 10
while (a LE b)
  a = a + b
```

```
                    // a => static[0]
                    // b => static[1]

                    push constant 20 // a = -20
                    neg
                    pop static 0
                    push constant 10 // b = -10
                    pop static 1

label LOOP          push static 0    // a LE b
                    push static 1
                    gt
                    if-goto LOOP_END
                    push static 0    // a = a + b
                    push static 1
                    add
                    pop static 0
                    goto LOOP

label LOOP_END
```