15CSE 381 Computer Organization and Architecture

Lab 5

19th July 2019

1 Questions

1. Consider the Java-like code below. Translate the code into MIPS instructions as directly as possible. Run your code to make sure it has correct behavior for initial values of x=0, x=1, and x=12.

```
int x = 1; // use $t3 to keep track of x's value
int y = 0; // use $t4 to keep track of y's value
if (x == 0) {
y++;
} else if (x == 1) {
y--;
} else {
y = 100;
}
```

2. Consider the Java-like code below. Translate the code into MIPS instructions as directly as possible.

```
int a = 2; // use $t6 to keep track of a's value
int b = 10; // use $t7 to keep track of b's value
int m = 0; // use $t0 to keep track of m's value
while (a > 0) {
m += b;
a -= 1;
}
```

3. Consider the Java-like code below. Translate the code into MIPS instructions as directly as possible.

```
int[] arr = {11, 22, 33, 44};
arrlen = arr.length;
```

// translation of above is given as

```
data
arr: .word 10 22 15 40
end:
.text
.globl main
main:
la $s0, arr
# this instruction puts the base address of arr in $s0
la $s1, end
subu $s1, $s1, $s0
srl $s1, $s1, 2 # now $s1 = number of elements in the array. justify this ?
```

// complete the translation of the below...

```
int evensum = 0;
// use $t0 to keep track of evensum's value
for (int i=0; i<arrlen; i++) {
if (arr[i] & 1 == 0) { // what does this condition mean?
evensum += arr[i];
}
}
```