Handout # 7
(Lack of) Register Allocation

Code generated for $i+1 < j$

```
lw    $a0 20($fp) ; load i into $a0
sw    $a0 0($fp)) ; spill it
la    $a0 int_lit8 ; load 1 into $a0
jal   Any.clone ; copy it
lw    $t1 0($fp) ; unspill i
lw    $t1 12($t1) ; get its true value
lw    $t2 12($a0) ; get 1 into $t1
add   $t1 $t1 $t2 ; compute i+1
sw    $t1 12($a0) ; store it in object
sw    $a0 0($fp) ; spill result
lw    $a0 16($fp) ; get j into $a0
lw    $t1 0($fp) ; unspill i+1 object
lw    $t1 12($t1) ; get i+1 into $t1
lw    $t2 12($a0) ; get true value of j
la    $a0 boolean_lit1 ; maybe result true?
blt   $t1 $t2 L21 ; compare them
la    $a0 boolean_lit0 ; no, it’s false
```

L21:

Questions:

- Why is $i$ stored as soon as it is loaded?
- Why is the object for $i+1$ spilled?
- Why are we loading at offset 12 from int_lit8 when we know the answer is 1?