Handout # 6

Cool for MIPS R2000

<table>
<thead>
<tr>
<th>Offset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>Garbage Collector Tag</td>
</tr>
<tr>
<td>0</td>
<td>Class tag</td>
</tr>
<tr>
<td>4</td>
<td>Object size (in 32-bit words)</td>
</tr>
<tr>
<td>8</td>
<td>Dispatch pointer</td>
</tr>
<tr>
<td>12</td>
<td>Attributes</td>
</tr>
</tbody>
</table>

Object Representation

<table>
<thead>
<tr>
<th>Offset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Any.Any</td>
</tr>
<tr>
<td>4</td>
<td>XXX.toString</td>
</tr>
<tr>
<td>8</td>
<td>XXX.equals</td>
</tr>
<tr>
<td>12</td>
<td>(other methods)</td>
</tr>
</tbody>
</table>

Dispatch Table Representation

Useful Instructions

The SPIM manual lists many instructions, but these are the ones you will need to know for this class. Here $Rdst$ and $Rsrc$ are registers: one of $v0$, $v1$, $a0$, $a1$, $a2$, $t0$, $t1$, $t2$, $sp$ or $ra$; $Src$ is either a register or an integer literal.

Arithmetic  Used for integer operations:

```
add $Rdst$, $Rsrc$, $Src2$
sub $Rdst$, $Rsrc$, $Src2$
mul $Rdst$, $Rsrc$, $Src2$
div $Rdst$, $Rsrc$, $Src2$
eg $Rdst$, $Rsrc$
```

Branch  Here the condition $cc$ can be one of eq, ne, lt, le, gt, ge:

```
b label
bcc $Rsrc$, $Src2$, label
```

Call Related

```
jal label
jalr $Rsrc$
jr $ra$
addiu $sp$, $sp$, ±4$n$
```

known procedure call
indirect call through register
return from procedure
pop/push space on stack

Data Movement

```
la $Rdst$, label
li $Rdst$, n
lw $Rdst$, 4$n$(Rsrc)
move $Rdst$, $Src$
sw $Rsrc$, 4$n$(Rsrc2)
```

load address of label into register
load literal value into register
load from memory at offset from register
store to memory at offset from register

Spring 2018