Java Virtual Machine instructions

*iload_0* integer load of local variable in slot 0
It pushes (creates) a "memory cell" at the top of the stack and copies the value of that local variable into that cell.

*ifne 6* if not equal
(1) Tests if the cell on top of the stack contains 0
(2) In all cases, pops (removes) the top stack cell.
(3) If not equal to 0, makes JVM continue at ("go to") instructions from position 6.
If equal to 0, "fall through" to the next instruction.

*icost_0* integer constant 0 (icost_1, icost_2 are similar)
Push a cell containing the value 0 on top of the stack

*ireturn* integer return
Do the return "dance" (Return to caller)
The return value, an integer, should be on top of the stack.

*ificmpne 13* if integer compare not equal
(1) Tests if the top two stack cells contain equal ints.
(2) In all cases, pop the top two stack cells.
(3) If not equal, go to instruction at pos. 6 and beyond.
(4) If equal, "fall through"