

JVM0 Instruction reference

<c> = constant

<i> = local variable index

<o> = branch offset

<f> = function address

// operand stack and locals

```
0x00 nop          S  -> S
0x59 dup          S,v -> S,v,v
0x57 pop          S,v -> S
0x5F swap        S,v1,v2 -> S,v2,v1
0x10 bipush <c>  S  -> S,c
0x15 iload <i>   S  -> S,V[i]
0x36 istore <i> S,v -> S; V[i] = v;
0x84 iinc <i>,<c>
```

// arithmetic

```
0x60 iadd        S,x,y -> S,x+y
0x64 isub        S,x,y -> S,x-y
0x68 imul        S,x,y -> S,x*y
0x6C idiv        S,x,y -> S,x/y
0x70 irem        S,x,y -> S,x%y
0x74 ineg        S,x  -> S,-x
0x78 ishl        S,x,y -> S,x<<y
0x7a ishr        S,x,y -> S,x>>y
0x7C iushr       S,x,y -> S,x>>>y
0x7E iand        S,x,y -> S,x&y
0x80 ior         S,x,y -> S,x|y
0x82 ixor        S,x,y -> S,x^y
```

// control

```
0xB1 ireturn    .,x -> .
0xA7 goto <o>
0x9F if_icmpeq <o> S,x,y -> S
0xA0 if_icmpne <o> S,x,y -> S
0xA1 if_icmplt <o> S,x,y -> S
0xA2 if_icmpge <o> S,x,y -> S
0xA3 if_icmpgt <o> S,x,y -> S
0xA4 if_icmple <o> S,x,y -> S
```

```
0xB8 invokestatic <f> S,v1,...,vn -> S,f(v1,...,vn)
```

```
int isqrt (int n) {
    int i = 0; int k = 0;
    while (0 <= k && k <= n) {
        k += 2*i + 1;
        i++;
    }
    return i-1;
}
```