Ex. logical error

if (cond)
    std;
    std; // executed unconditionally

Ex. Syntax error

if (cond)
    std;
    std;
else
    std;
    std;
for metavting:

if (cond) stmt;
else stmt;

or

if (cond) stmt1; else stmt2;

are sometimes OK

Ex. Sort3.java
Sort4.java

Exercise: Write Sort5.java
        and Sort6.java
more formatting:

```
Ex: if (cond1)
    if (cond 2)
        stmt1
    else
        stmt2
else
    stmt3
```

**Rule:** else is always paired with the most recent if.

another common construct:

```
if (cond1)
    s1
else
    if (cond 2)
        s2
    else
        if (cond 3)
            s3
        else
            s4
```
a better format:

```java
if (cond1)
  s1;
else if (cond2)
  s2;
else if (cond3)
  s3;
else
  s4;
```

Ex: `Compare1.java`

`Compare2.java`

Exercise: Write `Compare3.java`
Recall: ternary conditions.

\[
\text{cond ? exp1 : exp2}
\]

**Ex.**

```
int x, y = 6;
x = ((y > 5) ? 10 : 20);  // x gets 10
```

Equivalent to

```
int x, y = 6;
if (y > 5)
  x = 10;
else
  y = 20;
```
We can also 'cascade' this as:

```
Ex: int x, y = 6;
    x = (y < 5) ? 10 : (y < 10) ? 20 : (y < 15) ? 30 : 40;
```

Iterative control structures (loops)

- while
- for
- do-while

General form for while:

```
while (cond) { repetition cond (LRC)
    stmt;
    stmt;
    stmt;
    stmt; loop body
} stmt;
```
common while loop:

initialize LCV

\[ i = 0; \]

while \( i < n \)

// do stuff

\[ i++; \]

// increment LCV

? equivalent for loop:

\[ for \ (i=0; \ i<n; \ i++) \]

// do stuff

}\n
Ex Square2.java
General form for do-while:

do {
    start;
    start;
    start;
    start;
    loop body
    LRC
    while (condition);
    start;
    start;
    start;
}

Note: While loop body may execute 0 times, do-while body must execute at least once.

Ex. Squares3.java
Ex. logical error

```java
i = 0;
while (i < 10); { // stuff
    i++;
}
```

Ex. logical error

```java
for (i = 0; i < 10; i++); { // stuff
    
}
```

Ex. `Average.java`

Ex. `Max.java`
Exercise:

re-write both examples using:

- for loops
- do-while loops