

The conditional statements we will study are

- if
- if-else

There is another conditional construct we will study later called switch.

```

if (condition)
    stmt1;
    stmt2;

```

```

or
if (condition) {
    stmt1;
    :
    stmtK;
}
stmt(K+1);

```

"Condition" here is a Boolean expression. If condition is true then stmt1 is executed, followed by stmt2; If condition is false, stmt1 is skipped, then stmt2 is executed.

stmt1 is known as the TRUE BRANCH of the conditional.

THE TRUE BRANCH MAY CONSIST OF A COLLECTION OF ONE OR MORE STATEMENTS SURROUNDED BY BRACES { ... }, THIS IS KNOWN AS A COMPOUND STATEMENT OR BLOCK.

Ex

```
int a = 6, b = 5, temp;
if (a > b) {
    temp = a;
    a = b;
    b = temp;
}
System.out.println(a + " " + b);
```

~~THE~~ IF STATEMENT CAN HAVE AN OPTIONAL ELSE CLAUSE.

```
if (condition)
    stmt1;
else
    stmt2;
stmt3;
```

or

```

if (condition) {
    stmt1;
    ;
    stmtk;
} else {
    stmt(k+1);
    ;
    stmt(k+l);
}
stmt(k+l+1);
;

```

IF CONDITION IS TRUE, stmt1 is EXECUTED, stmt2 is SKIPPED, THEN EXECUTION RESUMES AT stmt3. IF CONDITION IS FALSE, stmt1 is SKIPPED, stmt2 is EXECUTED, FOLLOWED BY stmt3.

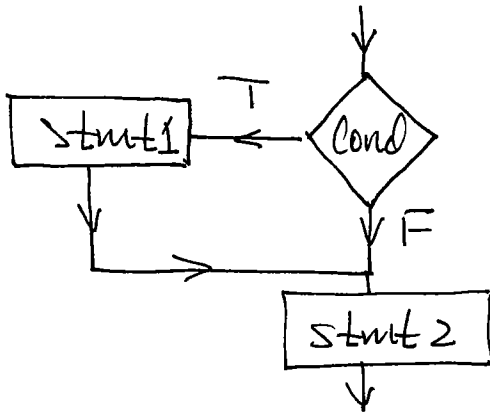
IN THIS CASE stmt1 is THE TRUE BRANCH AND stmt2 is THE FALSE BRANCH OF THE CONDITIONAL. EITHER ONE (OR BOTH) OF THESE BRANCHES CAN BE A COMPOUND STATEMENT, AS IN THE SECOND EXAMPLE.

Ex

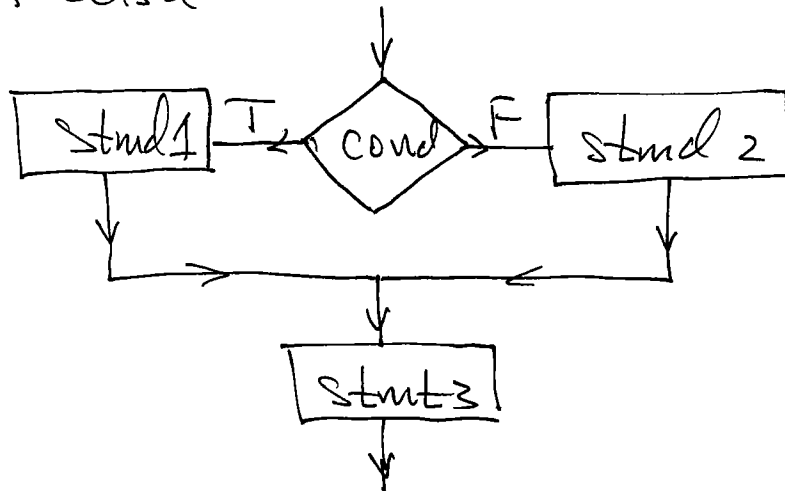
```
int a = 5, b = 6;
if (a < b)
    System.out.println(a + " " + b);
else
    System.out.println(b + " " + a);
```

These conditional statements can be pictorially as control flow diagrams

• if



• if-else



BRACES ARE NECESSARY WHEN EITHER BRANCH IS A COMPOUND STATEMENT.

Ex. logical error

```

if (cond)
    stmt1;
    stmt2; // stmt2 is UNCONDITIONAL
    stmt3 // IN SPITE OF INDENTATIONS

```

Ex SYNTAX ERROR

```

if (cond)
    stmt1;
    stmt2;
else // ONCE stmt2 is ENCOUNTED
    stmt3; // THE OPTION FOR AN else
           // clause is GONE

```

REMEMBER, INDENTATION IS MEANINGLESS TO THE COMPILER. IF EITHER BRANCH IS PARTICULARLY SHORT, IT IS ACCEPTABLE STYLE TO PLACE THEM ON SAME LINE AS IF (AND ELSE).

Ex

```
if (cond) stmt1;
else stmt2;
```

You can even put both on a single line

Ex.

```
if (cond) S1; else S2;
```

The following example is from lab3.

Ex

```
Scanner sc = new Scanner(System.in);
double x, y, q;
System.out.println("Enter numerator and denominator!");
if (sc.hasNextDouble()) x = sc.nextDouble();
else throw new RuntimeException("...");
if (sc.hasNextDouble()) y = sc.nextDouble();
else throw new RuntimeException("...");
if (y == 0) throw new RuntimeException("...");
q = x/y;
System.out.println("Quotient: " + q);
```

CONSIDER:

```

if (cond 1)
    if (cond 2)
        S1;
    else
        S2;
S3;

```

WHAT HAPPENS IF COND 1 IS FALSE?
WHICH IT DOES THE ELSE GO WITH?

ANSWER: ALWAYS THE MOST RECENT.

CONSIDER:

```

if (cond 1)
    S1;
else
    if (cond 2)
        S2;
    else
        if (cond 3)
            S3;
        else
            S4;

```

This happens frequently enough
 that we have a special style
 for it

```

if (cond 1)
    S1;
else if (cond 2)
    S2;
else if (cond 3)
    S3;
else
    S4;
  
```

ES SORT 3 NUMBERS :

```

int a, b, c, temp;
if (a > b) {
    temp = a;
    a = b;
    b = temp;
}
if (b > c) {
    ...
}
if (a > b) {
    ...
}
  
```


Ex

```
String s;
int m = 6;
```

```
System.out.print("time is " + m + " minute");
```

```
if (m == 1)
```

```
    s = ".";
```

```
else
```

```
    s = "s.";
```

```
System.out.println(s)
```

Ex.

```
(m == 1) ? "." : "s."
```

GENERAL FORM:

```
(condition) ? (expression1) : (expression2)
```

If condition is true, the value of the above expression is expression1, while if false, the value is expression2.

Ex. PRINTS ABSOLUTE VALUE OF DIFFERENCE

```
int a, b, c;
```

```
    // get a, b from user
```

```
c = (a < b) ? (b - a) : (a - b); // c = |a - b|
System.out.println(c);
```

Ex. Correctly pluralize "minute(s)" .

```
String s;
```

```
int m = 6; // change to m = 1
```

```
s = (m == 1) ? " " : "s.";
```

```
System.out.println("The time is " + m + "minute" + s);
```