

4. Flow Control

Lesson 1: Selective statements

4. Flow Control

- **Control flow** refers to the order in which statements are executed in an algorithm.
- Usually, an algorithm executes **sequentially**; that is the first statement executes, then the second and so on. However, it is often useful to be able to alter this flow.
- A **flow control statement** is a statement that changes the order of execution of subsequent statements.
- We will be talking about 3 types of flow control statements
 - Selection Statements
 - Iterative Statements
 - Jump Statements

4.1. Selection Statements

- A Selection Statement, evaluates some expression, and depending on the expression's value, selects one of several possible sets of statements to execute.
- In java, we have two types of selection statements:
 - If
 - Switch

4.1. Selection Statements

4.1.1. If Statements

- An If Statement has the form:

```
if (<Boolean expression>
{
    <statement1.1>;
    <statement1.2>;
    ...
}
else
{
    <statement2.1>;
    <statement2.2>;
    ...
}
```

- If **<Boolean expression>** is true, then **<statement1.1>**, **<statement1.2>** etc. are executed; else **<statement2.1>**, **<statement2.2>** etc. are executed. The “else” part is optional if there is nothing to be executed if **<Boolean expression>** is false.

4.1. Selection Statements

4.1.1. If Statements (cont...)

- If only one statement needs to be executed, the curly brackets can be omitted.

```
if (<Boolean expression>
    <statement1.1>;
else
    <statement2.1>;
```

4.1. Selection Statements

4.1.1. If Statements (cont...)

- If statements can be nested inside other if statements to give multiple if statements.

```
if (<Boolean expression1>
    <statement1.1>;
else if (<Boolean expression2>
    <statement2.1>;
else if (<Boolean expression3>
    <statement3.1>;
else
    <statement4.1>;
```

4.1. Selection Statements

4.1.2. Switch Statements

- An Switch Statement has the form:

```
switch (<Expression>
{
    case <value 1>:
        <statement1.1>;
        <statement1.2>;
    case <value 2>:
        <statement2.1>;
        <statement2.2>;
    case <value 3>:
        <statement3.1>;
        <statement3.2>;
    case <value 4>:
        <statement4.1>;
        <statement4.2>;
    ...
    default:
        <statementD.1>;
        <statementD.2>;
}
```

- The expression **<Expression>** must evaluate to a byte, int or a char.

4.1. Selection Statements

4.1.2. Switch Statements (cont...)

- **<Expression>** is compared with each of the case values **<value 1>**, **<value 2>**, **<value 3>** etc.
- If a match is found, the statements that correspond to the first match, and all the other case statements further down are executed
 - For example if **<Expression>** is equal to **<value 2>**, **<statement2.1>**, **<statement2.2>** etc. are executed; then, **<statement3.1>**, **<statement3.2>** etc. are executed, and so on.
- If a match is not found, default statements **<statementD.1>**, **<statementD.2>** etc. are executed.
- The default statements are optional. If it is not there, case statement completes without doing anything.