POWERING POTENTIAL

Getting started with conditionals

CONTENTS

- **Objectives**
  - To start covering the conditional expressions of the Java language
- **Contents**
  - Conditional control statements (if / else)
    - Adding ‘else if’ statement(s)
  - Relational operators and boolean expressions
  - Structuring if / else
  - Ordering of tests
  - Nesting if statements
  - More issues
  - `switch` statement
- **Hands on Labs**

Introducing ‘if’

```
if ( boolean_expression )
{
  statement(s);
}
```

Use Code Snippet

‘if’ Ctrl-Space in eclipse

'If' double tab in Visual Studio

```
String season = getSeasonDesc();

if ( season.equals("Winter"))
{
  // snowball time
}
```

Introducing ‘if else’

```
if ( boolean_expression ) {
  statement(s); // One or many lines
}
else {
  statement(s); // One or many lines
}
```

```
if ( boolean_expression )
  statement; // One line only
else
  statement; // One line only
```

```
int age = getAge();

if ( age < 18) {
  // code for when under 18 time
}
else {
 // code for when 18 or over
}
```

Introducing ‘else if’(s)

```
if ( boolean_expression )
{
   statement(s);
}
else if( boolean_expression )
{
   statement(s);
}
else if( boolean_expression )
{
   statement(s);
}
else
{
   statement(s);
}
```

‘if’ must come first

as many ‘else if’(s) as needed can come between them

‘else’ (if needed) comes last

'else if' example

```
String season = getSeasonDesc();
if ( season.equals("Winter")){
  // snowball time
}
// no code allowed here
else if ( season.equals("Spring")) {
  // weddings & ceremonies
}
else if ( season.equals("Summer")) {
  // long evenings
}
else {
  print("Leaves are falling");
}
print("This always happens");
```

Only 1 of these 4 'blocks' will ever run

'if's are normally related to each other, but don't have to be

```
it is Autumn!, no need for if..
```

Unconditional

Relational Operators to compare values

- **Produces a boolean value (`true/false`), never an int**

```
int year = getYear();

if ( year < 2000) {
  print("Seems ages ago");
}
if ( year = 2025 ) {
        // code…
}
if ( year == 2025 ) {
  print("The big one");
}
if ( year != 2025) {
        // code…
}
```

ERROR: assignment not equality

>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
==	equal to
!=	not equal to

An alternative

- **Both sample code achieve the same result but which one do you prefer?**

```
int age = getAge();

boolean isOver18 = (age >= 18);
```

```
int age = getAge();

boolean isOver18;

if ( age >= 18) {
   isOver18 = true;
}
else {
    isOver18 = false;
}
```

Structuring if / else .. identical code?

A

```
int i = calc();
if ( i == 0 )
{
   statement(s)1;
}
else
{
   ... ✓
   if ( i > 0 )
   {
      statement(s)2;
   }
   else
   {
      statement(s)3;
   }
   ... ✓
}
statement(s)4;
```

B

```
int i = calc();
if ( i == 0 )
{
   statement(s)1;
}
... ✗
else if ( i > 0 )
{
   statement(s)2;
}
... ✗
else
{
   statement(s)3;
}



statement(s)4;
```

C

```
int i = calc();
if ( i == 0 )
{
   statement(s)1;
}
✗
else if ( i > 0 )
{
   statement(s)2;
}
✗
else if ( i < 0 )
{
   statement(s)3;
}



statement(s)4;
```

On the motorway ..consider this code

A

```
int speed = calcAveSpeed();
if (speed > 70 )
{
   // been speeding
}
else if (speed <= 70 )
{
   // moving well
}
else if (speed <= 50 )
{
   // getting held up
}
else if (speed <= 30 ) {
   // slow progress
}
```

B

```
int speed = calcAveSpeed();
if (speed <= 30 )
{
   // slow progress
}
else if (speed <= 50 )
{
   // getting held up
}
else if (speed <= 70 )
{
   // moving well
}
else { // speed gt 70
   // been speeding
}
```

Last 3 ifs are in an incorrect sequence

..and when aveSpeed is -5?

Also be careful not to type ‘<‘ when you mean ‘<=‘ and vice versa

Still on the motorway ..

A

```
int speed = calcAveSpeed();
if (speed <= 50 ) {
  if (speed <= 30 ) {
    // slow progress
  }
  else {
    // getting held up
  }
}
else {
  if (speed > 70 ) {
    // been speeding
  }
  else {
    // moving well !!!
  }
}
```

We don't like nested if's, if we can avoid them

B

```
int speed = calcAveSpeed();
if (speed <= 30 )
{
   // slow progress
}
else if (speed <= 50 )
{
   // getting held up
}
else if (speed <= 70 )
{
   // moving well
}
else { // speed gt 70
   // been speeding
}
```

So much more readable and maintainable

More ‘if / else’ issues

A

```
int x = 12, y = 8;
if ( x >= 0 )
   if ( y >= 10) {
     print("x + y >= 10");
   }
else {
  print(“x is negative”);
}
```

Any problem here?

B

```
int x = -5;
if( x >= 0 );
{
  processPositiveValue(x);
}
processAnyValue(x);
```

What about here?

The switch Statement

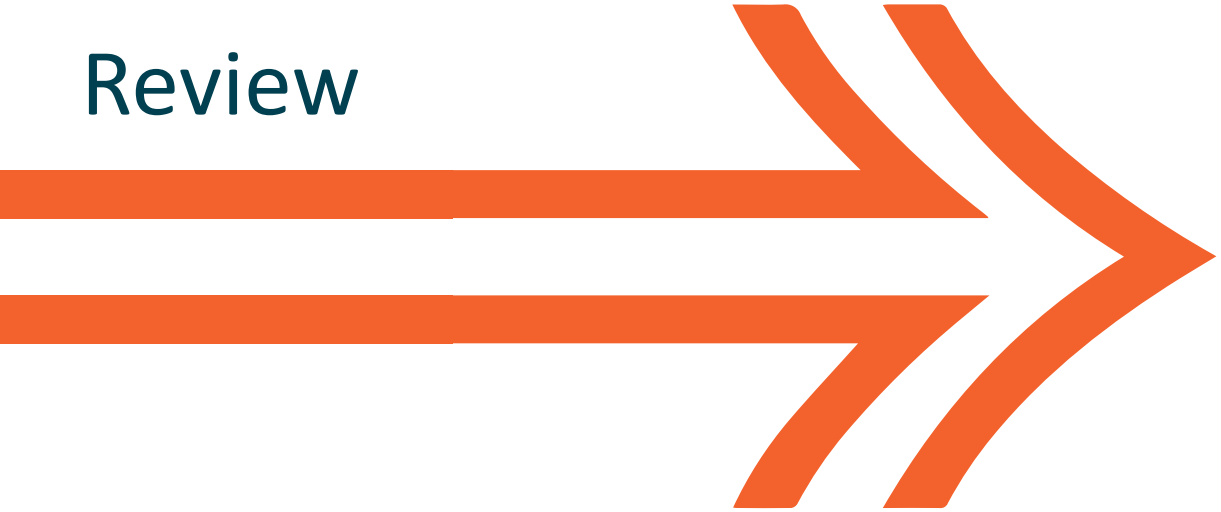
- **switch expression tests an integer, enum, char or String**
- **Statements may be in any order**
- **Often elegant alternative to**

```
if …
else if …
else if …
else …
```

```
switch ( score ) {
 case 0:
  res = "Clean sheet";
  break;
 case 1: case 2:
  res = "Not bad";
  break;
 default:
  res = "Leaky defence";
  break;
}
```

- **Java: BEWARE of accidentally dropping through to next section**
  - compiler does not force a ‘break’ statement

Review



- **Conditional statements if ...else if...else**
- **Relational operators**
- **Structuring code and ordering of tests**
- **`switch` statement**

Hands On Labs

- **Kid In a Candy Store**