List of Slides

Title

Java Just in Time

John Latham

October 8, 2018

Chapter 4

Conditional execution

Chapter aims

- Not every part of a program need be **execute**d every time.
  - Some parts only under certain **condition**s.
- We meet **if statement**s and **if else statement**s.

Section 2

Example:
Oldest spouse 1

Aim

AIM: To introduce the idea of **conditional execution**, implemented by the **if else statement**, and controlled by **boolean expression**s based on the use of **relational operator**s.

Execution: conditional execution

- Not sufficient to always obey all instructions every **run**
  - some problems need **conditional execution**
    - **execute** some bits **condition**ally:
    - depending on values of **variable**s.

Expression: boolean

- An **expression** which **evaluate**s to **`true`** or **`false`**
  - a **condition**
  - also called a **boolean expression**.
- Used for controlling **conditional execution**.

Expression: boolean: relational operators

- Six **relational operator**s for comparing values to make up **condition**s
  - show us **relation**s between items in pairs of values
  - **binary infix operator**s. . .
  - yield **`true`** or **`false`**.

Expression: boolean: relational operators

Operator	**Title**	**Description**
`==`	Equal	This is the **equal** operator, which provides the notion of **equality**. `a == b` yields **`true`** if and only if the value of `a` is the same as the value of `b`.
`!=`	Not equal	This is the **not equal** operator, providing the the notion of not equality. `a != b` yields **`true`** if and only if the value of `a` is *not* the same as the value of `b`.
`<`	Less than	This is the **less than** operator. `a < b` yields **`true`** if and only if the value of `a` is less than the value of `b`.

Expression: boolean: relational operators

Operator	Title	Description
>	Greater than	This is the **greater than** operator. `a > b` yields **`true`** if and only if the value of `a` is greater than the value of `b`.
<=	Less than or equal	This is the **less than or equal** operator. `a <= b` yields **`true`** if and only if the value of `a` is less than value of `b`, or is equal to it.
>=	Greater than or equal	This is the **greater than or equal** operator. `a >= b` yields **`true`** if and only if the value of `a` is greater than value of `b`, or is equal to it.

Oldest spouse 1

Coffee time: Which of the following **conditions** are true?

a)	`5 < 4`	b)	`5 < 5`	c)	`5 < 6`
d)	`5 > 4`	e)	`5 > 5`	f)	`5 > 6`
g)	`5 <= 4`	h)	`5 <= 5`	i)	`5 <= 6`
j)	`5 >= 4`	k)	`5 >= 5`	l)	`5 >= 6`
m)	`5 == 4`	n)	`5 == 5`	o)	`5 == 6`
p)	`5 != 4`	q)	`5 != 5`	r)	`5 != 6`
s)	`5 > 4 == 9 < 15`	t)	`5 < 4 != 9 > 15`		

Do the last two examples above suggest that `==` and `!=` have a different **operator precedence** from the other **relational operators**? If so, are they higher or lower?

Statement: if else statement

- The **if else statement** allows **conditional execution.**
- Three parts
  - **condition** / **boolean expression**
  - **true part** – **statement** for when condition is **`true`**
  - **false part** – **statement** for when condition is **`false`**
- Syntax:
  - **reserved word `if`**
  - condition in brackets
  - true part
  - **reserved word `else`**
  - false part

Statement: if else statement

- E.g. assuming **variable** `noOfPeopleToInviteToTheStreetParty`

```
if (noOfPeopleToInviteToTheStreetParty > 100)
  System.out.println("We will need a big sound system!");
else
  System.out.println("We should be okay with a normal HiFi.");
```

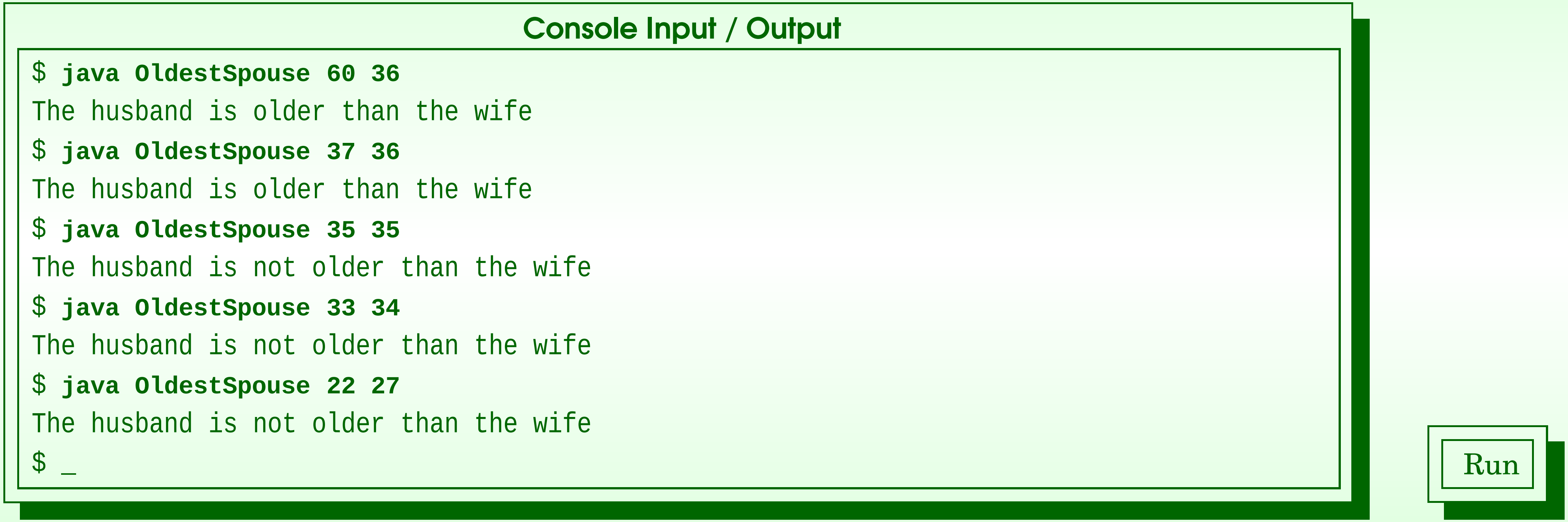
- Get one message or the other depending on `noOfPeopleToInviteToTheStreetParty` – never both.
- Notice brackets, semi-colons and **indentation**.

Oldest spouse 1

```java
public class OldestSpouse
{
  public static void main(String[] args)
  {
    int husbandsAge = Integer.parseInt(args[0]);
    int wifesAge = Integer.parseInt(args[1]);

    if (husbandsAge > wifesAge)
      System.out.println("The husband is older than the wife");
    else
      System.out.println("The husband is not older than the wife");
  }
}
```

Trying it



```
$ java OldestSpouse 60 36
The husband is older than the wife
$ java OldestSpouse 37 36
The husband is older than the wife
$ java OldestSpouse 35 35
The husband is not older than the wife
$ java OldestSpouse 33 34
The husband is not older than the wife
$ java OldestSpouse 22 27
The husband is not older than the wife
$ _
```

Coursework: `MaxTwoDoubles`

(Summary only)

Write a program to find the maximum of two given numbers, using an **if else statement**.

Section 3

Example:
Oldest spouse 2

Aim

AIM: To introduce the idea of nesting **if else statement**s.

Statement: if else statement: nested

- Can have **if else statement nested** inside another.
  - i.e. **true part** and/or **false part** is another if else statement.
- E.g. Depending on `noOfPeopleToInviteToTheStreetParty` get one of *three* messages.

```
if (noOfPeopleToInviteToTheStreetParty > 300)
  System.out.println("We will need a Mega master 500 Watt amplifier!");
else
  if (noOfPeopleToInviteToTheStreetParty > 100)
    System.out.println("We will need a Maxi Master 150 Watt amplifier!");
  else
    System.out.println("We should be okay with a normal HiFi.");
```

- Notice **indentation** – following usual nesting rule, but. . . .

Statement: if else statement: nested

- Make exception to usual rule for if else statements nested in false parts:

```
if (noOfPeopleToInviteToTheStreetParty > 300)
  System.out.println("We will need a Mega master 500 Watt amplifier!");
else if (noOfPeopleToInviteToTheStreetParty > 100)
  System.out.println("We will need a Maxi Master 150 Watt amplifier!");
else
  System.out.println("We should be okay with a normal HiFi.");
```

- Reflects abstraction: is one selection offering one choice from three.
- Don't have to increase indentation if using many nested if else statements – as long as nested in false parts.

Oldest spouse 2

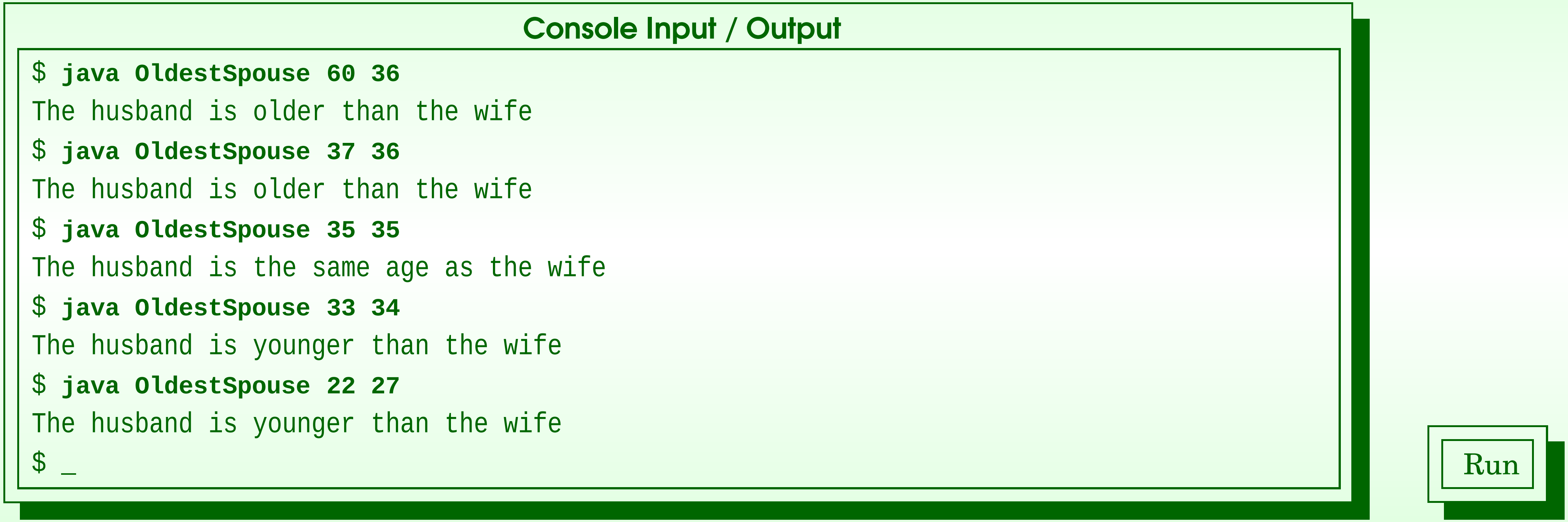
Coffee time: If we wanted some program code to **execute** one out of N choices, how many **if else statements** would we need?

Oldest spouse 2

```java
public class OldestSpouse
{
  public static void main(String[] args)
  {
    int husbandsAge = Integer.parseInt(args[0]);
    int wifesAge = Integer.parseInt(args[1]);

    if (husbandsAge > wifesAge)
      System.out.println("The husband is older than the wife");
    else if (husbandsAge == wifesAge)
      System.out.println("The husband is the same age as the wife");
    else
      System.out.println("The husband is younger than the wife");
  }
}
```

Trying it



```
$ java OldestSpouse 60 36
The husband is older than the wife
$ java OldestSpouse 37 36
The husband is older than the wife
$ java OldestSpouse 35 35
The husband is the same age as the wife
$ java OldestSpouse 33 34
The husband is younger than the wife
$ java OldestSpouse 22 27
The husband is younger than the wife
$ _
```

Trying it

Coffee time: What do you think of the following code as an alternative to the **if else statement**s we wrote in our program? Would it work? Is it as easy to read?

```
if (husbandsAge <= wifesAge)
  if (husbandsAge != wifesAge)
    System.out.println("The husband is younger than the wife");
  else
    System.out.println("The husband is the same age as the wife");
else
  System.out.println("The husband is older than the wife");
```

Trying it

Coffee time: What about the following code instead? What makes it better than the code above?

```
if (husbandsAge < wifesAge)
  System.out.println("The husband is younger than the wife");
else if (husbandsAge == wifesAge)
  System.out.println("The husband is the same age as the wife");
else
  System.out.println("The husband is older than the wife");
```

Trying it

Coffee time: And finally, what about the following code? What makes it *less* good than the one above?

```
if (husbandsAge == wifesAge)
  System.out.println("The husband is the same age as the wife");
else if (husbandsAge < wifesAge)
  System.out.println("The husband is younger than the wife");
else
  System.out.println("The husband is older than the wife");
```

Coursework: `DegreeCategory`

(Summary only)

Write a program to report the degree category of a given mark.

Section 4

Example:
Film certificate age checking

Aim

AIM: To introduce the **if statement** without a **false part**.

Statement: if statement

- Sometimes want nothing done if **condition** is **`false`**.
- E.g. could have empty **false part**:

```
if (noOfPeopleToInviteToTheStreetParty > 500)
  System.out.println("You may need an entertainment license!");
else ;
```

  - An **empty statement** between **`else`** and semi-colon!

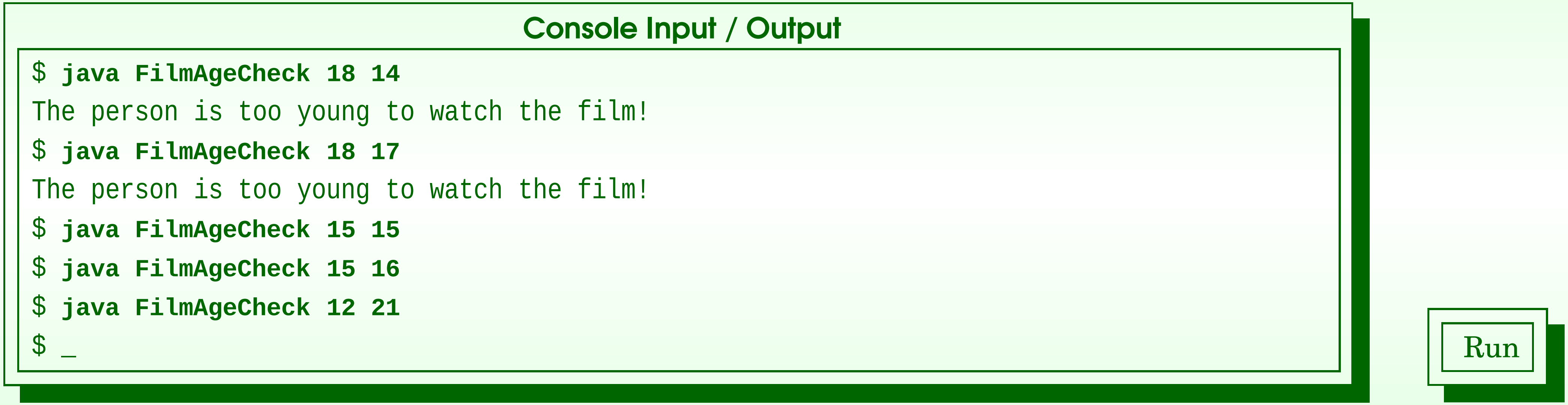
- Instead can use **if statement** – has no **`else`**, nor false part.

```
if (noOfPeopleToInviteToTheStreetParty > 500)
  System.out.println("You may need an entertainment license!");
```

Film certificate age checking

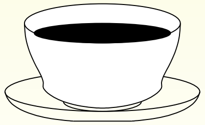
```java
public class FilmAgeCheck
{
  public static void main(String[] args)
  {
    int minimumAge = Integer.parseInt(args[0]);
    int ageOfPerson = Integer.parseInt(args[1]);
    if (ageOfPerson < minimumAge)
      System.out.println("The person is too young to watch the film!");
  }
}
```

Trying it



```
$ java FilmAgeCheck 18 14
The person is too young to watch the film!
$ java FilmAgeCheck 18 17
The person is too young to watch the film!
$ java FilmAgeCheck 15 15
$ java FilmAgeCheck 15 16
$ java FilmAgeCheck 12 21
$ _
```

Trying it



Coffee time: Under what **condition** will the following code print the message?

```
if (noOfPeopleToInviteToTheStreetParty > 500);
  System.out.println("You may need an entertainment license!");
```

(Hint: count the semi-colons.)

Trying it

Coffee time: When do we get the quiet party message with this code?

```
if (noOfPeopleToInviteToTheStreetParty > 100)
  if (noOfPeopleToInviteToTheStreetParty > 500)
    System.out.println("You may need an entertainment license!");
else
  System.out.println("It will be a fairly quiet party.");
```

(Hint: which **`if`** does the **`else`** match?)

Coursework: `PassFailDistinction`

(Summary only)

Write a program to report the pass or fail status of an exam candidate, giving a message of distinction if appropriate using an **if statement**.

Section 5

Example:
Absolute difference

Aim

AIM: To introduce the **compound statement**.

Statement: compound statement

- The **compound statement** – list of **statement**s between { and }.
  - E.g. body of a **method**?
- Semantics – **sequential execution**.
- Most useful with statements containing one other statement
  - but want more than one.
- E.g. conditionally get *three* messages:

```
if (noOfPeopleToInviteToTheStreetParty > 500)
{
  System.out.println("You may need an entertainment license!");
  System.out.println("Also hire some street cleaners for the next day?");
  System.out.println("You should consider a bulk discount on lemonade!");
}
```

- Note: no indent between if statement and compound, only within it.

Statement: compound statement

- Less useful, compound can be empty:

```
if (noOfPeopleToInviteToTheStreetParty > 500)
{
  System.out.println("You may need an entertainment license!");
  System.out.println("Also hire some street cleaners for the next day?");
  System.out.println("You should consider a bulk discount on lemonade!");
}
else {}
```

Absolute difference

```java
public class AbsoluteDifference
{
  public static void main(String[] args)
  {
    double firstNumber = Double.parseDouble(args[0]);
    double secondNumber = Double.parseDouble(args[1]);

    double absoluteDifference;

    if (firstNumber > secondNumber)
    {
      System.out.println("The first number is larger than the second");
      absoluteDifference = firstNumber - secondNumber;
    }
```

Absolute difference

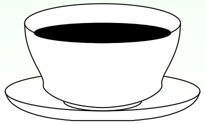
```
    else if (firstNumber == secondNumber)
    {
      System.out.println("The two numbers are equal");
      absoluteDifference = 0;
    }
    else
    {
      System.out.println("The second number is larger than the first");
      absoluteDifference = secondNumber - firstNumber;
    }
    System.out.println("The absolute difference is " + absoluteDifference);
  }
}
```

Trying it

Console Input / Output

```
$ java AbsoluteDifference 123.4 123.45
The second number is larger than the first
The absolute difference is 0.04999999999999716
$ _
```

Run



Coffee time: Are you surprised by the inaccuracy of the above result?

Console Input / Output

```
$ java AbsoluteDifference 123.45 123.45
The two numbers are equal
The absolute difference is 0.0
$ java AbsoluteDifference 123.45 123.4
The first number is larger than the second
The absolute difference is 0.04999999999999716
$ _
```

Run

Trying it

Note: **equality** of **real**s is a dangerous notion. . . .

Console Input / Output

```
$ java AbsoluteDifference 123.45 123.4500000000000001
The two numbers are equal
The absolute difference is 0.0
$ _
```

Run

Concepts covered in this chapter

- Each book chapter ends with a list of concepts covered in it.
- Each concept has with it
  - a self-test question,
  - and a page reference to where it was covered.
- Please use these to check your understanding before we start the next chapter.