Ian Kenny

October 2, 2017

Software Workshop

Lecture 2a: Coding example.

In this lecture

Program 1

- Prompts the user for two numbers.
- Outputs which of the numbers is largest, or that they are equal if that is the case.

Program 2

- Accepts two numbers from the command line.
- Outputs which of the numbers is largest, or that they are equal if that is the case.

Starting out

What type of things will we need to do?

Starting the program

We need a class. We will call this one MaximumOf2KB ('KB' for keyboard version). We will need a main method too.

```java
public class MaximumOf2KBa {

    public static void main(String[] args) {

    }
}
```

Listing 1 : MaximumOf2KBa.java

What next?

What shall we do next?

Create the Scanner object

Create the Scanner object that we need to read the keyboard. This requires an `import` statement.

```java
import java.util.Scanner;

public class MaximumOf2KBb {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

    }
}
```

Listing 2 : MaximumOf2KBb.java

What next?

What about prompting the user?

Prompt the user

```java
import java.util.Scanner;

public class MaximumOf2KBc {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.println("Enter two integers separated
            by a space.");
    }
}
```

Listing 3 : MaximumOf2KBc.java

What next?

What shall we do next?

Read the keyboard input

```java
import java.util.Scanner;

public class MaximumOf2KBd {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.println("Enter two integers separated
            by a space.");

        String inputLine = in.nextLine();

    }
}
```

Listing 4 : MaximumOf2KBd.java

What next?

Now the real work begins.

What do we need to do with the input string?

Splitting the string

We need to split the input string into two parts each representing one of the input numbers. If the input is "234 456", for example, we need two strings: one that is equal to "234" and another that is equal to "456".

We can do this by finding the space character in the input string because that is the *delimiter*, i.e. the character that signifies where the two parts of the string can be divided[1].

---

[1]Incidentally, the delimiter doesn't have to be a space character. It could be another character or more than one character.

Finding the space

```java
import java.util.Scanner;

public class MaximumOf2KBe {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.println("Enter two integers separated
            by a space.");

        String inputLine = in.nextLine();

        int spacePosition = inputLine.indexOf(' ');
    }
}
```

Listing 5 : MaximumOf2KBe.java

What next?

What do we need to do next?

Splitting the string

We actually split the string using the `substring()` method of the `String` class.

The first use of the method will give us the substring *up to* the space. The second use will give us the rest of the string *after* the space.

Splitting the strings

```java
import java.util.Scanner;

public class MaximumOf2KBf {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.println("Enter two integers separated by a space.");

        String inputLine = in.nextLine();

        int spacePosition = inputLine.indexOf(' ');

        String val1 = inputLine.substring(0, spacePosition);

        String val2 = inputLine.substring(spacePosition + 1);

    }
}
```

Listing 6 : MaximumOf2KBf.java

What next?

What do we need to do next?

Converting to integers

Now we need to convert the strings we have created, which contain the two input numbers, into integers. Currently, of course, they are still strings.

We do this by using a method from the `Integer` class called `parseInt()`. This method takes a `String` as a parameter and returns its integer value. Hence the string "123" will be converted to the integer 123.

Note that any string other than one representing a positive or negative integer will cause an error.

Converting to integers

```java
import java.util.Scanner;

public class MaximumOf2KBg {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.println("Enter two integers separated by a space.");

        String inputLine = in.nextLine();

        int spacePosition = inputLine.indexOf(' ');

        String val1 = inputLine.substring(0, spacePosition);

        String val2 = inputLine.substring(spacePosition + 1);

        int num1 = Integer.parseInt(val1);
        int num2 = Integer.parseInt(val2);
    }
}
```

Listing 7 : MaximumOf2KBg.java

What next?

What do we need to do next?

What next?

Finally, we have two numbers we can compare, which is the whole point of the program. Now all we need to do is create an if-else statement of some kind to compare the values.

Look again at the requirements ...

What might this if-else construct look like?

Comparing the numbers

```java
import java.util.Scanner;

public class MaximumOf2KBh {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.println("Enter two integers separated by a space.");

        String inputLine = in.nextLine();

        int spacePosition = inputLine.indexOf(' ');

        String val1 = inputLine.substring(0, spacePosition);

        String val2 = inputLine.substring(spacePosition + 1);

        int num1 = Integer.parseInt(val1);
        int num2 = Integer.parseInt(val2);

        if (num1 == num2)
            System.out.println("The numbers are the same.");
        else if (num1 > num2)
            System.out.println("The first number was largest.");
        else System.out.println("The second number was largest.");

    }
}
```

Listing 8 : MaximumOf2KBh.java

How to create the command line version?

Look at the program again. How can we convert it to a command line version?

Command line version

```java
public class MaximumOf2CL {

    public static void main(String[] args) {

        /* note that the space in the input means the user
           would have to use double quote marks to
           surround the two input numbers.
        E.g. "234 567" */
        String input = args[0];

        int spacePosition = input.indexOf(' ');

        String val1 = input.substring(0, spacePosition);

        String val2 = input.substring(spacePosition + 1) ;

        int num1 = Integer.parseInt(val1);
        int num2 = Integer.parseInt(val2);

        if (num1 == num2)
            System.out.println("The numbers are the
                same.");
        else if (num1 > num2)
            System.out.println("The first number was
                largest.");
        else System.out.println("The second number was
                largest.");
    }
}
```

Listing 9 : MaximumOf2CL.java