Tutorial 7: Methods

This tutorial will provide practice at implementing methods, passing parameters to them and returning results from them.

Task 1: BMI calculator

Write a program to calculate a person's BMI based on the height and weight input.

The `main()` method should:

- call a method `heightInInches()` to input and return the user's height
- call a method `weightInPounds()` to input and return the user's weight
- call a method `outputBMI()` to calculate and output the user's BMI

The `heightInInches()` method should:

- input 2 integers: the user's height in feet (in the range 2 – 7) and inches (in the range 0 – 11)
- validate the input and prompt again until the data is valid
- return the user's height in inches (12 inches in a foot)

The `weightInPounds()` method should:

- input 2 integers: the user's weight in stone (in the range 3 – 30) and pounds (in the range 0 – 13)
- validate the input and prompt again until the data is valid
- return the user's weight in pounds (14 pounds in a stone)

The `outputBMI()` method should:

- receive 2 parameters: `heightInInches` and `weightInPounds`
- calculate and output the BMI using the formula:

$$\text{BMI} = \text{weight} * 703 / \text{height}^2$$

Step 1: Analyse the methods

- using the above, determine the method header for the 3 methods (return type, name and parameters required) and store it in a file called `BMIAnalysis.doc`

Step 2: Create a NetBeans project

- create a new project called `BMIproj` in a folder called `T7`

Step 3: Write source code

- add a new file called `BMI` to the `BMIproj` project
- using the analysis from step 1, encode the solution containing 3 methods

Step 4: Test your program and take screen shots

- run your program with a height of 5' 10'' and weight of 11 stone , 6 pounds
- take a screen shot of the output and save it in your project folder as `BMI.jpg`

Portfolio requirements

- The NetBeans project for this completed task
- `BMIAnalysis.doc` from step 1 containing details of the method headers
- `BMI.jpg` from step 3, containing a screen shot of the output when the user inputs 5 foot 10, 11 stone 6

Task 2: Smallest4

Write a method called `smaller()` that returns the smaller of the 2 integers passed to it as parameters.

Write a program that prompts the user to input 4 integers and outputs the smallest of the 4 values. The `main()` method may only contain input and output statements and calls to the `smaller()` method.

Step 1: Create a NetBeans project

- create a new project called `Smallest4Proj` in a folder called `T7`

Step 2: Write source code

- add a new file called `Smallest4` to the `Smallest4Proj` project
- encode the `smaller()` method and the program that uses that method to output the smallest of 4 integers

Step 3: Test your program

- run your program
- take a screen shot of the output and save it in your project folder called `Smallest4.jpg`

Portfolio requirements

- The NetBeans project for this completed task
- `Smallest4.jpg` from step 3, containing a screen shot of the output

Task 3: Factorial

The factorial of a number `n`, written as `n!`, is the product of all positive integers less than or equal to `n`.

Thus, `4! = 4 x 3 x 2 x 1 = 24`.

Write an application that prompts the user to input a positive integer greater than 1 and outputs the factorial of that number. Write a method that calculates the factorial using an iteration, and returns the answer to the `main()` method.

Step 1: Create a NetBeans project

- create a new project called `FactorialProj` in a folder called `T7`

Step 2: Write source code

- add a new file called `Factorial` to the `FactorialProj` project
- encode the `calculateFactorial()` method and the program that uses that method to prompt the user for a number and output the factorial of that number

Step 3: Test your program

- run your program
- take a screen shot of the output and save it in your project folder called `Factorial.jpg`

Portfolio requirements

- The NetBeans project for this completed task
- `Factorial.jpg` from step 3, containing a screen shot of the output

Task 4: Recursive Factorial

This task is to do exactly the same as Task 3, but instead of using an iteration in the `calculateFactorial()` method, you are to make it a recursive method.

Although you might need to do some research into recursion, do not copy the answer for this task from any source.

Portfolio requirements

- The NetBeans project for this completed task
- `RecursiveFactorial.jpg` containing a screen shot of the output