

## Revision Exercises

*Before you get into algorithms, you need to remember all the skills you built last year. These include user input, conditional logic, loops and method creation.*

### Loops

#### Exercise 1.1)

Write a program that:

- Repeatedly takes in numbers from the user. Input should terminate when the user enters -1.
- Once -1 has been entered, the program should print out the sum and average of the entered numbers

#### Exercise 1.2)

Write a program that:

- Asks the user for two numbers between 1 and 10 (width and height)
- Prints a square of asterisks of the user-specified size. (Hint: use nested for loops)
- Example:
  - If the user enters 4 and 5, the program should display:

```
****
****
****
****
****
```

#### Exercise 1.3)

Write a program that:

- Takes in two numbers from the user.
- Adds up all the numbers between those the user entered.
- Outputs the result.
- Example: If a user enters 3 and 7, then the program should print out 25 (i.e. 3+4+5+6+7).  
**Your program should still work correctly, even if the user enters the bigger number first (i.e. 7 and then 3).**

## Writing static Methods

#### Exercise 2.1)

Write a static method called `getValidInteger()` that:

- Takes in a String as a parameter
- Uses this text as prompt to the user.
- Confirms the data being entered is a number.
  - If the data is a number, it should be stored.
  - If it is not a number, the user should be informed their entered information was inappropriate, then prompted to enter a value again (using the same text as before).
  - The value entered should be confirmed as a number. If it is not, this process should repeat.
- Returns the entered (and validated) number

Your method should be written in `CalcMax.java`, and should be a static method.

#### Exercise 2.2)

Write a static method called `findMax()` that:

- Takes in three ints as parameters
- Returns the largest of the three numbers
- Note: You may not use any built-in methods from Java to accomplish this.

Your method should be written in `CalcMax.java`, and should be a static method.

### Exercise 2.3)

Write a program that:

- Takes in 3 numbers from the user
  - You should use your `getValidInteger` method to validate each of the numbers as they are entered
- Displays the highest of the entered numbers.
  - You should use your `findMax()` method to decide which number should be displayed.

### Exercise 2.4)

Write a static method called `findMax()` that:

- Takes in an array of ints as a parameter
- Returns the largest number in the array
- Note: You may not use any built-in methods from Java to accomplish this.

Your method should be written in `CalcMax.java`, and should be a static method. This is an example of an OVERLOADED method, as there are multiple methods with the same name, but each has a different set of parameters.

### Exercise 2.5)

Write a static method called `getValidInteger()` that:

- Takes in a String prompt and two ints representing the upper and lower limits of allowable values
- Uses the text as a prompt to the user.
- Confirms the data being entered is a number, and that it is within the specified range.
  - If the data is a number AND within the specified range, it should be stored.
  - If it is not a number, the user should be informed their entered information was inappropriate, then prompted to enter a value again (using the same text as before).
  - If it is a number but is not within the specified range, the user should be informed that their value is outside the allowable range, then prompted to enter an appropriate value (using the same text prompt as before)
  - The new value entered should also be confirmed as a number and within the specified range. If it is not, this process should repeat.
- Returns the entered (and validated) number

Your method should be written in `CalcMax.java`, and should be a static method. This is another example of an OVERLOADED method, as there are multiple methods with the same name, but each has a different set of parameters.

## Utility Classes

### 3.1)

Copy your `getValidInteger()` methods to a class called `InputUtility.java`.

### 3.2)

Modify your answer to exercise 2.3 to use the `getValidInteger()` method from your `InputUtility`.