Programming 1

Tutorial 5

# Review Exercises: Arrays, Loops & Conditionals

## 1. Write programs with loops that compute:

1. The sum of all even numbers between 2 and 100 (inclusive).
2. The sum of all squares between 1 and 100 (inclusive).
3. All powers of 2 from 20 up to 220.

1 2 4 8 16 32 64 128 256

1. The sum of all odd numbers between a and b (inclusive), where a and b are inputs.

Scanner -> get a and b -> loop from a to b -> check odd numbers, calculate sum

1. The sum of all odd digits of an input (for example, if the input is 32677, the sum would be 3 + 7 + 7 = 17).

Extract each digit -> check for odd digits -> add to sum

**2.** Write programs that read a sequence of integer inputs (you decide the size of the sequence) and print:

1. The smallest and largest of the inputs.
2. The number of even and odd inputs.
3. Cumulative totals. For example, if the input is 1 7 2 9, the program should print 1 8 10 19.
4. All adjacent duplicates. For example, if the input is 1 3 3 4 5 5 6 6 6 2, the program should print 3 5 6.

**3.** Write a program that prints a multiplication table, like this:

1 2 3 4 5 6 7 8 9 10

2 4 6 8 10 12 14 16 18 20

3 6 9 12 15 18 21 24 27 30

. . .

10 20 30 40 50 60 70 80 90 100

(Advanced) Make it look like so:

1 2 3 4 5 6 7 8 9 10

2 4 6 8 10 12 14 16 18 20

3 6 9 12 15 18 21 24 27 30

. . .

10 20 30 40 50 60 70 80 90 100

**4.** Write a program that reads an integer and displays, using asterisks, a filled and hollow square, placed next to each other. For example, if the side length is 5, the program should display:

\*\*\*\*\* \*\*\*\*\*

\*\*\*\*\* \* \*

\*\*\*\*\* \* \*

\*\*\*\*\* \* \*

\*\*\*\*\* \*\*\*\*\*

# Activity 1

Write a program that reads an integer and output its binary form. The program should make use of a static method called toBinary(...).

int -> toBinary() -> String

reason: binary is much longer than decimal

public static String toBinary(int n)

# Activity 2

Write programs that read a line of input as a string and print:

1. Only the uppercase letters in the string.
2. Every second letter of the string.
3. The string, with all vowels replaced by an underscore.
4. The number of vowels in the string.
5. The positions of all vowels in the string.

# Activity 3

Write a program that reads a set of floating-point values. Ask the user to enter the values (prompting only a single time for the values), then print:

1. The average of the values.
2. The smallest of the values.
3. The largest of the values.
4. The range, that is the difference between the smallest and largest.

# Activity 4

Write a program to let user enter a credit card number which may contain digits and dashes, such as: 4123-56 78-9012 3450. Remove all spaces and dashes to make it only contain digits (or better yet, remove everything that is non-digit). The result should be: 4123567890123450.

Then, check that the credit card number is valid. A valid credit card number will yield a result divisible by 10 when you:

Calculate the sum of all digits. Add to that sum every second digit, starting with the second digit from the right. Then add the number of digits in the second step that are greater than four. The result should be divisible by 10.

For example, consider the card number **4012 8888 8888 1881**. The sum of all digits is 89. The sum of the red-colored digits is 46. There are 5 colored digits which are larger than 4, so the result is 140. 140 is divisible by 10 so the card number is valid.

Some credit card numbers for you to test:

3782-8224-6310 005

3714-4963-5398 431

3787-3449-3671 000

5555 5555 5555 4444

5105-1051-0510-5100

5019 7170 1010 3742

6331-1019-9999-0016

# Activity 5

Write a program to ask the user to enter an integer which must belong to the range [10, 20]. Your program should ask the user to enter again (and again) if the user enters an invalid value.

## Expected result:

Enter an integer between 10 and 20: 6

But you must enter an integer between 10 and 20: 3

But you must enter an integer between 10 and 20: 30

But you must enter an integer between 10 and 20: 15

Thank you, 15 it is!

## Instructions

This should be done with a while loop.