

Tutorial 3

4COSC010C Software Development II

Loops

Exercise 1: Loops 1:

Write a program that uses a loop to print the following:

```
1
2
3
4
5
```

Exercise 2: Loops 2:

Extend the previous program (exercise 3) to print the following using two loops (nested loop):

```
1
22
333
4444
55555
```

Hint: You can use `System.out.print("\n")` to print a new line.

Exercise 3: Loops with a step.

Modify again the previous program (exercise 2) to print the following output:

```
22
4444
666666
88888888
```

Exercise 4: Positive Integers

Write a java program using a while loop to accept an positive integer n and a letter from the user to print the letter n times.

Enter a number : 5

Enter a letter : Y

output : YYYYY

Exercise 5: The Fibonacci series

The Fibonacci series is as follows:

1, 1, 2, 3, 5, 8, 13, 21, 34

Can you see how each number is generated? After the first two numbers, each number is the sum of the previous two numbers. This series is unusual in that it can be used to describe natural phenomena, such as angular displacement of plant leaves for maximizing incident light.

Write a program to compute this series to the n^{th} term.

Exercise 6: Passcode.

Write a program that checks if a passcode is correct. The user has 4 attempts to input the passcode correctly. The passcode is 486251. To read an int, you can use `input.nextInt()`. If the user enters the passcode correctly should display a message saying "Correct passcode" and the program should end.

Note: Think about which control structure you will use to allow the user inserting a maximum of 4 passcodes. Do not repeat your code 4 times. Will you use an if, a for loop, a while loop, a switch? Think about the most adequate control structure.

Exercise 7: Challenge: Secret number.

Write a program that generates a random number between 1 and 20 and asks the user to guess the number. The user should be able to enter a new number if the number is incorrect.

Note: Use Random to generate a random number. To use Random, you will have to import it first:
`import java.util.Random.`

Read below source to learn more about random numbers

<https://www.javatpoint.com/how-to-generate-random-number-in-java>

Exercise 8: Armstrong numbers.

Write a program that prints all Armstrong numbers between 1 and 500. An Armstrong number is a number that the sum of cubes of each digit is equal to the number itself.

Example:

$153 = (1*1*1) + (5*5*5) + (3*3*3).$

Hint: To get the remainder of a division, you can use the symbol %.

Homework questions

1. See the program below.

Number to the power '*power*' is $\text{number} * \text{number} * \text{number} \dots$ (*power* number of times)

This program multiplies number by number while the counter is less than power.

Why doesn't it work? How do we debug?

```
import java.util.*;
import java.io.*;
public class Main {
    public static void main(String[] args)
    {

        int number, power, count;
        int total = 0;
        Scanner input = new Scanner(System.in);
        System.out.println("Enter number : ");
        number = input.nextInt();
        System.out.println("Enter power ");
        number = input.nextInt();
        count = 0;
        while (count <= power )
        {
            total = number * number;
        }
        System.out.println(" the answer is " + total);
    }
}
```

2. Factorial 5 ($5!$) = $5 \times 4 \times 3 \times 2 \times 1$. The program below asks for the factorial number. Dry run it. Try entering in a 4 for the number. Why doesn't the loop work? Put a print statement to see how many times the loop goes around. fix the below code so that it prints the factorial.

```
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        int number, count, factorial;
        int total = 0;
        Scanner input = new Scanner(System.in);
        System.out.println("Enter number ");
        number = input.nextInt();
        count = 0;
        while (count > number )
        {
            number = count;
            factorial = total * number;
        }
        System.out.println("the factorial is " + total);
    }
}
```

3. Given any number, we can create a new number by **adding the sums of squares of digits of that number**. For example, given 203, our new number is $4 + 0 + 9 = 13$. If we repeat this process, we get a sequence of numbers:

$n \rightarrow 13 \rightarrow 10 \rightarrow 1 \rightarrow 1$

Sometimes, like with 203, the sequence reaches (and stays at) 1. Numbers like this are called happy. Not all numbers are happy. If we started with 11, the sequence would be:

$11 \rightarrow 2 \rightarrow 4 \rightarrow 16 \rightarrow \dots$

This sequence will never reach 1, and so the number 11 is called unhappy.

fix the code so that it prints the factorial.