Linnéuniversitetet
Kalmar Växjö

# Linnaeus University

## 1DV532 – Starting out with Java
## Assignment 1 Report

Student: Evan Huynh
Student ID: eh223im@student.lnu.se

# Table of Contents

*Report* ........ *3*

**Exercise 1: Print** ........ 3

**Exercise 2: Numbers** ........ 3

**Exercise 3: ISBN** ........ 3

**Exercise 4: ArmstrongNumber** ........ 3

**Exercise 5: Days** ........ 4

**Exercise 6: SquareRoot** ........ 4

**Exercise 7: Fibonacci** ........ 4

*Source code* ........ *5*

*Bibliography* ........ *8*

# Report

## Exercise 1: Print

The first exercise asks for a lot of **System.out.println("something")** so it is self-explanatory. To create 4 lines of text in the console output, we have to add the "\n" to every word, or if you want to bloat, create for lines of **System.out.println("something")**.

To create a square, it is better to draw it as shown in the source code. Every "\n" marks the new line.

## Exercise 2: Numbers

Since this one asks for user input, we are using the Scanner class with the System.in as a source. There is a check to make sure that user actually input a three digits number. If not, the program will simply exit after user make a mistake.

Since this is a three-digit number, we can get the first digit by divide it to 100, get the second by divide it by 10 then modulo the result 10, and then get the last digit by modulo 10. The rest of the task is just simply adding (or multiplying) every digit according to the requirement. Since there are only three digits, it is not necessary to add (or multiply) all the digits by the *for* loop.

## Exercise 3: ISBN

This exercise has the similar idea to the exercise 2, however there are 9 digits to process. The algorithm remains the same as we can get the last digit by of 9 digits. Since we also count the number of 0 in the input and we ask users for the integer, we assume that padding 0 to the beginning is a better solution.

To obtain the last digit, we simply modulo the input by 10. To obtain the next digit, we divide the number by 10 and then continue to mod it by 10. Repeat until the input is 0.

To obtain the number of digits in an integer, we use the formula $d = \lfloor \log_{10} a \rfloor + 1$, for $a > 0$, otherwise $d = 1$ (if $a = 0$) [1]. Assuming user enters an integer less than 9 digits, we can then apply the ISBN multiplication backward, starting from 9 to the last digit and 1 to the first digit. Taking the last digit works the same as in exercise 2.

After that, we modulo the result to 11 and print as many 0 as needed for the missing 0, then add X if the congruence is 10, otherwise output the modulo result.

## Exercise 4: ArmstrongNumber

There are three steps to determine if a number is Armstrong:

1. Count number of digits
2. Sum the power of individual digit to the number of digits, doing backward from the last digit
3. Output the result

If user wants to repeat the program, simply ask for letter Y at the first position of the string, otherwise we just exit the program. There is also a check if the starting range is larger than the ending range.

## Exercise 5: Days

By modular arithmetic and a lot of if statements, we should be able to know what day in the week it should be. Since there are 7 days in a week, modulo 7 is enough.

We also add a check to ensure that every result is larger or equal 0 to simplify the problem.

## Exercise 6: SquareRoot

The square root exercise is just making a guess then comparing percentage difference with the old guess until it is less than 1%. Doing this within a do…while or while loop is both viable. However, in this example we use do…while loop.

For making the guess, we can use the exact description of how to guess, then update the guess as required.

After that printing out the result using print and prinf statements.

## Exercise 7: Fibonacci

Let a, b, c be integers. To compute the Fibonacci sequence, at first, we set a to 0, b and c to 1, then:

1. Set a = b
2. Set b = c
3. Set c = a + b
4. Repeat the processes until c>=1000

Therefore, the algorithm fulfills the definition of Fibonacci's sequences:

$$\begin{cases} F_1 = 1 \\ F_2 = 1 \\ F_n = F_{n-1} + F_{n-2} \end{cases}$$

To check for the oddness of the number, check if the modulo 2 returns 1.

## Source code

Here is my source code for all the exercise.

1. Print

```
package eh223im_assign1;

public class Print {

    public static void main(String[] args) {
       System.out.println("Write once, run everywhere!");

       System.out.println();

        System.out.println("Write\nonce\nrun\neverywhere");

        System.out.println();

        System.out.println("******************************\n" +
                           "* Write once, run everywhere! *\n"+
                           "******************************\n" );
    }
}
```

2. Numbers

```
package eh223im_assign1;

import java.util.Scanner;

public class Numbers {
    public static void main(String[] args) {
        int a;
        int b;
        int c;
        Scanner s = new Scanner(System.in);
        // Assuming no check
        System.out.print("Enter a three-digit integer number: ");
        a=s.nextInt();
        if (a>=100 && a<=999) {
            b = (a / 100) + ((a / 10) % 10) + (a % 10);
            System.out.println("Sum of the digits of the integer number is: " + b + ".");
            c = (a / 100) * ((a / 10) % 10) * (a % 10);
            System.out.println("Product of the digits of the integer number is: " + c + ".");
        }
    }
}
```

3. ISBN

```
package eh223im_assign1;

import java.util.Scanner;

public class ISBN {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int b=0;
        System.out.print("Enter the first 9 digits of an ISBN as integer: ");
        int a = s.nextInt();
        int c = a;
        if (a>=0 && a<=999999999) {
            int d;
            if (c == 0) {
                d = 1;
            } else {
                d = (int) (Math.floor(Math.log(c) / Math.log(10)) + 1);
            }
            for (int i = 0; i < d; i++) {
                int lastdigit = a % 10;
                a = a / 10;
                b += lastdigit * (9 - i);
            }
            b = b % 11;
            System.out.print("The ISBN-10 number is: ");
            for (int i = 0; i<9-d; i++) {
                System.out.print(0);
            }
            if (b == 10) {
                System.out.println(c + "" + "X");
            } else {
                System.out.println(c + "" + b);
            }
        }
    }
}
```

4. ArmstrongNumber

```
package eh223im_assign1;

import java.util.Scanner;

public class ArmstrongNumber {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        char a = 'y';
        while (a=='y') {
            System.out.print("Enter the starting number of the range: ");
            int b = s.nextInt();
            System.out.print("Enter the ending number of the range: ");
            int c = s.nextInt();
            if (c >=0 && c>=b) {
                for (int i = b; i <= c; i++) {
                    if (isArmstrong(i)) {
                        System.out.println(i);
                    }
                }
            }
            System.out.print("Do you want to repeat? (Y/N): ");
            a=s.next().toLowerCase().charAt(0);
            System.out.println();
        }
    }

    static boolean isArmstrong(int c) {
        int lastdigit;
        int counter=0;
        int sum=0;
        int a=c;
        if (c == 0) {
            counter = 1;
        } else {
            counter = (int) (Math.floor(Math.log(a) / Math.log(10)) + 1);
        }
        while(a!=0) {
            lastdigit=a%10;
            a=a/10;
            sum+=Math.pow(lastdigit,counter);
        }
        return sum==c;
    }
}
```

5. Days

```
package eh223im_assign1;

import java.util.Scanner;

public class Days {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter today's day: ");
        int a=s.nextInt();
        System.out.print("Enter the number for future day: ");
        int b=s.nextInt();
        if (a>=0 && a+b>=0) {
            System.out.print("Today is " + main2(a % 7) + " and the fu-
ture day is " + main2((a + b) % 7));
        }
    }

    static String main2(int c) {
        if (c==0) {
            return"Sunday";
        } else if (c==1) {
            return"Monday";
        } else if (c==2) {
            return"Tuesday";
        } else if (c==3) {
            return"Wednesday";
        } else if (c==4) {
            return"Thursday";
        } else if (c==5) {
            return"Friday";
        } else if (c==6) {
            return"Saturday";
        } else return "Unknown";
    }
}
```

6. SquareRoot

```
package eh223im_assign1;

import java.util.Scanner;

public class SquareRoot {
    public static void main(String[] args) {
        System.out.print("Enter an integer to estimate the square root
of: ");
        Scanner scan = new Scanner(System.in);
        int a = scan.nextInt();
        // Source for the idea: https://stackoverflow.com/questions/3144610/integer-division-how-do-you-produce-a-double
        if (a > 0) {
            double guess = a * 1.0 / 2;
            double r = a * 1.0 / guess;
            double percent;

            System.out.println("Current guess: " + guess);
            do {
                double newguess = (guess + r) / 2;
                percent = (Math.abs(newguess - guess)) * 100;
                guess = newguess;
                r = a * 1.0 / guess;
                System.out.println("Current guess: " + guess);
            } while (!(percent <= 1.0));
            System.out.print("The estimated square root of " + a + " is ");

            System.out.printf("%.2f", guess);
            System.out.println();
        }
    }
}
```

7. Fibonacci

```
package eh223im_assign1;

public class Fibonacci {
    public static void main(String[] args) {
        int a = 0;
        int b = 0;
        int c = 1;
        int d = 0;

        while (c<1000) {
            System.out.println(c);
            if (c%2==1) {
                d+=c;
            }
            a=b;
            b=c;
            c=a+b;
        }
        System.out.println("Sum of all odd terms in the sequence: "+d);
    }
}
```

## Bibliography

[1] Brilliant.org, "Finding The Number of Digits," 16 June 2020. [Online]. Available: https://brilliant.org/wiki/finding-digits-of-a-number/.