



## 1DV532 – Assignment 1

- The assignment comprises **six exercises** and **passing at least four exercises is mandatory** to pass the assignment.
- Your submission will receive a grade from A to F where F is Fail.
- You are allowed to improve your work after the initial submission before the deadline.
- Passing grade is final; you will not get an opportunity to correct/improve after grading.
- Collaboration and discussions are encouraged, but each student must report **individually! You are not allowed to copy code from other students, books articles, blogs, wikis or any other source!** Each submission will pass through a plagiarism/clone detection system before correction. **If plagiarism is detected, the assignment will be failed and a formal investigation will be initiated.**

### Preliminaries (Basic steps to structure and organize your code)

#### 1. Install Java

Download and install Java SE JDK: [www.oracle.com/technetwork/java/javase/downloads](http://www.oracle.com/technetwork/java/javase/downloads). We recommend you to use the latest stable version. Also, there are plenty of instruction videos available in YouTube. Just search for "Install Java X" where X is your operating system.

#### 2. Install Eclipse (you may use any other IDE of your choice)

Download and install Eclipse IDE for Java

Developers: <https://www.eclipse.org/downloads/packages/release/2021-03/r/eclipse-ide-java-developers>. Once again, there are plenty of instruction videos available in YouTube. Just search for "Install Eclipse X" where X is your operating system.

Feel free to contact teaching assistants if you require a support or help in Java/Eclipse installation.

#### 3. Setup Eclipse Workspace

Before you start programming, do the following.

- Create an Eclipse *workspace* (a folder) with the name `java_courses` on some location in your home directory.
- Create a *Java project* with the name `1DV532` inside the workspace.
- Create a *package* with the name `YourLnuUserName_assign1` inside the project. For example, it might look something like `wo222ab_assign1`.
- Save all program files from the exercises in this assignment inside the package `YourLnuUserName_assign1`.
- In the future, create a new package (`YourLnuUserName_assignX`) for each assignment and a new project (with the course code as name) for each new course using Java.

## Exercises

- Write a program with a class name **Print.java** that prints the phrase *Write once, run everywhere!*
  - on one line,
  - on four lines, one word on each line,
  - inside a rectangle made up by the characters \*, such as shown

Following is an example output required:

```
Write once, run everywhere!
Write
once,
run
everywhere!
*****
* Write once, run everywhere!      *
*                                *
*****
```

- Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. Write a program with a class name **BMI.java** that prompts the user to enter a weight in pounds and height in inches and displays the BMI.

Note: One pound is 0.45359237 kilograms and one inch is 0.0254 meters.

Following is a sample run of the program:

```
Enter weight in pounds: 95.5
Enter height in inches: 50
BMI is 26.8573
```

- Write a program with a class name **Numbers.java** that asks the user to enter a single three-digit integer number, for instance, 123, 456, or 483. The program then calculates and prints, the sum and product of the three digits the integer consists of.

Below is an example of the program execution:

```
Enter a three-digit integer number: 483
Sum of the digits of the integer number is: 15.
Product of the digits of the integer number is: 96
```

Note 1: Spaces between the input values are not allowed.

Note 2: You are not allowed to use more than one integer type variable to hold the initial input value.

Hint: Think of using Modulus % and Division operators to get individual digits of the entered value.

4. Write a program with a class name **Grades.java** that prompts users to enter total marks and marks obtained for a student. The program then determines and prints whether a student has passed the course or not with a grade (A, B, C, D, E, or F) achieved by the student. To pass the course a student is required to score at least 60% marks. The grade is computed based on following criteria:

Grades criteria	
Obtained Marks	Grade
Less than 60%	F
60 - 69%	E
70 - 74%	D
75 - 79%	C
80 - 89%	B
>= 90%	A

Following are two example runs of the program:

**Example 1:**

```
Please enter total marks: 100
Please enter marks obtained: 72
Congratulations, you have passed the course with grade D
```

**Example 2:**

```
Please enter total marks: 100
Please enter marks obtained: 55
You could not pass this time :(, please try again! Your grade is F
```

**Note:** Displaying suitable error message for following problem instances will help to achieve better grades.

- User enters a non-integer or non-positive value for total marks
- User enters a non-integer, negative, or higher than total marks value for marks obtained.

5. Write a program with a class name **Payment.java** that computes the change a customer should receive when he has paid a certain sum. The program should exactly describe the minimum number of Swedish bills and coins that should be returned *rounded off* to nearest krona (kr).

**Example:**

Price: 372.38

Payment: 1000

Change: 628 kronor

1000kr bills: 0

500kr bills: 1

200kr bills: 0

100kr bills: 1

50kr bills: 0

20kr bills: 1

10kr coins: 0

5kr coins: 1

2kr coins: 1

1kr coins: 1

6. An ISBN-10 (International Standard Book Number) consists of 10 digits:  $d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}$ . The last digit,  $d_{10}$ , is a [checksum](#), which is calculated from the other nine digits using the following formula:

$$(d_1 * 1 + d_2 * 2 + d_3 * 3 + d_4 * 4 + d_5 * 5 + d_6 * 6 + d_7 * 7 + d_8 * 8 + d_9 * 9) \% 11$$

Note: % represents the modulus operator

If the checksum is 10, the last digit is denoted as X according to the ISBN-10 convention.

Write a program with a class name **ISBN.java** that prompts the user to enter the first 9 digits and displays the 10-digit ISBN (including leading zeros). Your program should read the input as a primitive type **integer**, String or any other type is not allowed to use for the input value. Below are two sample runs of the program:

Run 1:

```
Enter the first 9 digits of an ISBN as integer: 013601267
The ISBN-10 number is: 0136012671
```

Run 2:

```
Enter the first 9 digits of an ISBN as integer: 013031997
The ISBN-10 number is 013031997X
```