

Homework: Introduction to Java

This document defines homework assignments from the [“Java Basics” Course @ Software University](#). Please submit as homework a single **zip / rar / 7z** archive holding the solutions (source code) of all below described problems.

Problem 1. Play with Eclipse

Install **Java 8** and **Eclipse** for Java 8. Start Eclipse and play with it. Try to write some Java code. You do not have to submit anything in your homework for this problem.

Problem 2. Empty Java Project in Eclipse

Create a new **Java project in Eclipse** called “**Intro-Java-Homework**”. This project will hold all your homework files (Java source code). For each problem (exercise), add a separate Java class with self-descriptive name like “**Print-Hometown**” or “**Sum-Two-Numbers**”. You do not have to submit anything in your homework for this problem.

Problem 3. Print Your Hometown

Create a simple Java program **PrintHometown.java** (console application) to print the name of your hometown, compile and run it. Submit the source code as part of your homework.

Problem 4. Play with the Java API Documentation

Play with Java API Documentation. You may find it online at <http://docs.oracle.com/javase/8/docs/api/>.

- Find information about **System.out.println** method.
- Find information about the **LocalDateTime** class.
- Find information about the **java.util.Scanner** class.

You do not have to submit anything in your homework for this problem.

Problem 5. Print the Current Date and Time

Create a simple Java program **CurrentDateTime.java** to **print the current date and time**. Submit the Java class **CurrentDateTime** as part of your homework.

Problem 6. Sum Two Numbers

Write a program **SumTwoNumbers.java** that **enters two integers** from the console, calculates and prints their **sum**. Search in Internet to learn how to read numbers from the console. Examples:

a	b	result
5	2	8
12	-7	5

Problem 7. Console-Based Compilation

Using **javac** compile from the console the program **SumTwoNumbers.java**. Execute at the console the obtained class file **SumTwoNumbers.class** and check whether it works correctly. Submit the console commands you have used to compile and run the program as part of your homework in a text file **compile-and-run-commands.txt**.

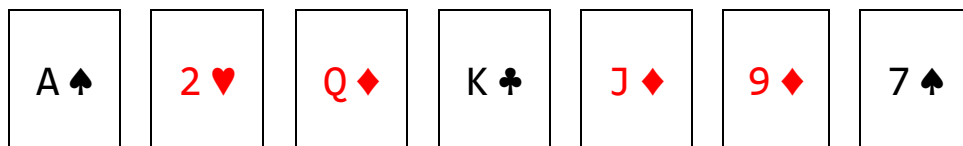
Problem 8. * Sort Array of Strings

Write a program that enters from the console **number n** and **n strings**, then sorts them alphabetically and prints them. Note: you might need to learn how to use loops and arrays in Java (search in Internet). Examples:

Input	Output
5 Sofia Plovdiv Varna Pleven Bourgas	Bourgas Pleven Plovdiv Sofia Varna

Problem 9. *** Generate a PDF by External Library

Write a program to **generate a PDF document** called **Deck-of-Cards.pdf** and print in it a **standard deck of 52 cards**, following one after another. Each card should be a rectangle holding its face and suit. A few examples are shown below:



You are free to choose the size of each card, the spacing between the cards, how many cards to put in each line, etc.

Note: you will need to use an external **Java library for creating PDF documents**. Find some in Internet. Put your JAR files in a folder called "**lib**" (this is a standard approach in Java projects) and reference them in the build path.

Hint: solve the problem step by step:

1. Generate the deck of cards and print it at the console (the console in Eclipse fully supports Unicode).
2. Find a Java library for generating PDF documents and play with it. Try to print some string in a PDF document.
3. Print the cards in PDF file (without the rectangular border).
4. Try to change the colors of the red cards.
5. Try to add the rectangular border around each card, e.g. by putting tables in the PDF.

Problem 10. JAR Archive + Console Execution

Create a **JAR archive** holding your code from the previous program. Write a script (**run.cmd** / **run.sh**) to run your program from the Windows / Linux console (depending of your environment). You should include in the classpath the JAR holding your code as well as the JARs holding the external libraries that your code uses. Submit the script along with the JAR files as part of your homework.

Problem 11. * Create a Windows EXE

Create a Windows executable file (**.exe**) from your JAR files. You may use some tool like Launch4J (<http://launch4j.sourceforge.net>).