

## Programming Languages 2

### Lesson 5

**Finish the project created last week, by learning how to read data from the keyboard.**

1. Read the name of a new Dish from the keyboard.  
(You can do this with the Scanner class. Instantiate a new scanner using `Scanner sc = new Scanner(System.in);`

Then use the `nextLine` method of it to read a String till “enter”).

2. Read the time of a new Dish from the keyboard.  
(You can do this using the same scanner as above. This time use the `nextInt()` method of Scanner.)
3. Add three ingredients of the Dish using `Scanner.next()`.
4. Finally read the recipe from the keyboard using `Scanner.nextLine()`;
5. Add the recipe to the RecipeBook that you create using `MyRecipeBook`;

**Create a project to record your spending**

1. Start a new NetBeans project. The name of the project is “Spending”. The name of the main class is “Main”.
2. Add a new class to the project named “Expense”. The class has the following attributes: `name`, `price` and `date`. The date is represented by three a `LocalDate` value. (See [java.time package JavaDoc](#) for more about representing date and time in Java)
3. Add getters and setters for all the attributes and a constructor that can set all of them.
4. Add an exception class to the project. The name of it is `InvalidDateException`.
5. Make the constructor of the `Expense` throw this exception if the given date is not valid (it is older than 100 years).
6. Add an empty file to the project root folder with the name `data.txt`.  
Fill it with the following data:  
`2013-11-15@book@34.50`  
`2013-10-11@apple@5.50`  
`2013-12-01@chair@110`
7. Create a new `List` in the main method and read all data from the file into it. (you may chose the `ArrayList` implementation of the `List` interface)
8. Print out the elements of the list. Each item has to be in a different line. To do this add a new static method to the main class that gets the `List` as a parameter and prints it out.
9. Make the `Expense` class implement thee `Comparable` interface. Implement the `compareTo` method. It returns a negative number if the given object is before the parameter and a positive one in the opposite case. It returns 0 if the two objects may be replaced.
10. Sort the list using the `sort()` method of the `Collections` class, and print out the list again.