In this first session, we introduce the main features and syntax of Java. In particular, we speak about the main advantages of Object Oriented Programming (OOP) and the reason why it represents a very important step in the direction of the progress of abstraction. We see the main concepts of objects and classes, and how they are constructed. Classes have fields and methods: we see how fields are defined and initialised and how methods are defined, implemented and called.

We then speak about primitives: a *special* kind of objects, including `int`, `float`, `double`, `char` and others, which are allocated in the `stack` and not on the `heap` (the `stack` is faster and more efficient, but less flexible). We see operators, aliasing and execution control.

So that you can have a look at the material covered today, the following is a list of the classes we see. It is written in the same order as presented and includes a short description of the corresponding topic presented.

- `session1.helloworld.HelloWorld`: very first example of Java class. We see how a class is defined, and the `main` method.
- Classes in `session1.oophelloworld`: we see a class, `Message`, with two methods and one field. Here we see how methods are defined and implemented, and how fields are defined and initialised. The class `OopHelloWorld` contains the `main` method. It shows a first example about how methods are called and fields manipulated.
- `session1.operators.Operators`: operators acting with primitives are shown and introduced. We have various examples of their use.
- Code in `session1.tank`: here we observe the phenomenon of `aliasing`: for objects which are not `primitives`, we manipulate their reference rather than their value. So when you assign `FirstObject = SecondObject`, they are allocated in the same piece of memory. In this classes we see the consequences of that.
- `session1.divisible.Divisible`: very simple example of `if/else` statement.
- Code in `session1.testval`: again an example `if/else` statement. The class `Comparison` has two methods that return a `boolean` value and one that returns an `int` value. The last one shows the use of the `ternary operator`.
- Code in `session1.oopdivisible`: also about the `if/else` statement, but this is a first exercise for you. In the `main` method of `MainDivisible`, you have to construct an object instance of `CheckDivisible`, give a value to its fields and call the appropriate method.
- `session1.randomvariable.WhileRandom`: here we start to see iterations, and in particular this is an example of `do..while` (or `while`, if we write it slightly differently). Important to note the Java `Random()` class: we will use it quite often.
- `session1.elevator.Elevator` very easy example of `for` loop.
- Code in `session1.gauss`: example of the use of the `for` loop: the class `SumOfIntegers` has a method which computes the sum of the first n natural numbers, printing them as well.
- Code in `session1.primenumbers`: exercise for you, on the `for` loop and `if` statement: you have to write a class with a method to check if a number is prime.
- `session1.switches.ARandomSwitch`: example of the `switch` statement.