Lesson 3

If you find any open questions in some of the presented exercises then feel free to make your own decisions. However, your decisions should be reasonable and must not contradict any of the conditions explicitly written for the exercise. Please, write comments in the programs that clarify your assumptions/decisions, if any.

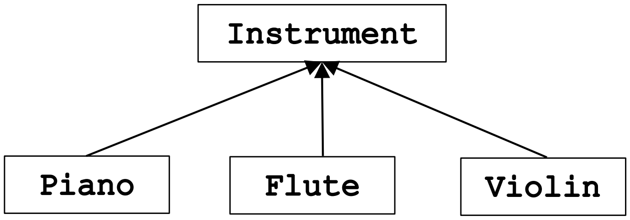
Exercise 1

Read lab 3 description, available from the course web site. Write the definitions of the classes in the expressions hierarchy, to be used for a classroom discussion. Full implementation of the classes is part of the lab exercise. Thus, you do not need to bring any implementation for the member functions to this lesson.

Exercise 2

Review the concept of dynamic binding and virtual functions introduced in Fö 11.

Consider the musical instruments hierarchy of classes given in the file **exerc2.cpp**. The hierarchy is depicted in the figure below.



The program in the file **exerc2.cpp** is ready to be compiled and executed.

Modify the classes' code such that the displayed output becomes as follows. Do not modify the function **tune** neither the **main**.

```
Piano::play with tune middleC
Flute::play with tune middleC
Violin::play with tune middleC
Piano::play with tune middle
```

Exercise 3

For this exercise you should review the concept of static members presented in Fö 9 and the code example in the folder **`Clock_static`** distributed with Fö 9.

Consider again the hierarchy of the previous exercise and the code given in the file **`exerc3.cpp`** (that compiles and executes).

Modify the code such that the given classes keep a list of all brands of musical instruments that exist (or have existed) and the current number of existing instruments of each brand. This list can only be updated by member functions of the classes forming the musical instrument's hierarchy. Use an array of 100 slots to represent the list of brands. Note that an instrument brand is a **`string`** like "**`Stradivarius`**".

Finally, add a member function **`display_brands()`** to display a table of instruments brands, as the one below, and call this function at the specific points indicated in the code (search for "*call **`display_brands()`***").

```
Instrument       Quantity
===========================
Yamaha               1
PureTone             2
Stradivarius         0
Bach                 1
```

The expected output of the program is given in the file **`ex3_out.txt`**.

Lycka till!!