

Programming in C/C++

Exercises 7

1a) Implement the following classes:

- class “StaticBase” with a static member function and a static int variable.
- class “PureBase” with a constructor, a member function and an int variable.
- class “VirtBase” with a constructor, a non-virtual member function, a virtual member function and an int variable.
- class “VirtDer” that inherits “public” from VirtBase with a non-virtual member function, a virtual member function of the same name as for VirtBase and an int variable.

All member functions have no parameters, return no value (i.e. void) and simply increment the int variable if called.

In main() define the following:

```
PureBase pure;
VirtBase vbase;
VirtDer vder;
PureBase *ppure = new PureBase();
VirtBase *pvbase = new VirtBase();
VirtDer *pvder = new VirtDer();
VirtBase *pvbaseder = pvder;
```

Now, create all function calls that are possible with these variables (it should be 14).

Group them according to similarities of the calls (e.g., all calls that only have an extra parameter) and add a comment describing the group.

Using the stopwatch of Exercise 5 measure the time needed for x repetitions of each call (i.e. create a loop around each call). To have meaningful results set $x > 10000000$.

Measure also the call of the static member function of StaticBase in the same way.

Compare and interpret the results.

Hints and Notes:

- Use a preprocessor macro to combine the start of the stopwatch, the loop, the call of an arbitrary function (given as parameter) and the stop of the stopwatch. Then, each test is only 1 line of code!
- To minimize the effects of other programs running on your computer, you should run the above process several times, for example by wrapping a loop around all tests. You do not need to calculate the averages in the program – use your favorite spreadsheet for it.
- Nevertheless, calls in the same group that should need approximately the same time may have different measurements. Try to rearrange the calls and rerun the program. You do not need to find a layout where all the calls result in correct timing (as it might be different on my platform anyway). Report on the differences between the groups and on possible outliers in your measurements.

(code 40pts, comments 15pts, report 10pts)

b) Use VirtBase, VirtDer and pvbaseder as defined before.

Implement and describe using comments two possibilities to call the non-virtual member function of VirtDer using pvbaseder (Hint: The == operator for the type_info object is overloaded as one is expecting). Measure the time needed for both versions. Report! Are there any disadvantages of the faster version compared to the slower one? If so, describe an example.

(code 15pts, comments 10pts, report 10pts)