



University of Antwerp
| Faculty of Science

C++ Tracing Exercises

Advanced Programming
2021-2022

Presentation Overview

1. What is the purpose of these exercises?
2. How to complete such an exercise?
3. How can you practice for the exam?
4. How will the exam exercise work?

Exercise purpose

What:

- Given the following code, what is the output?
- Expected output corresponds to the execution trace
- Not allowed to actually execute it

Need to know:

- What is being executed?
- When is it executed?

Simplified exercise example

Code:

```
int app_example()
{
    COMP_MISC_FUNCTION_TRACER;
    COMP_MISC_LOG_TRACER("statement: Engine e(10);");
    Engine e(10);
    return 0;
}
```

Output:

```
Starting main program ...
---> function body: int app_example()
    statement: Engine e(10);
    ---> member for object: 0x7fff15ad6840 : ODemo::Engine::Engine(double)
    <--- member for object: 0x7fff15ad6840 : ODemo::Engine::Engine(double)
    ---> member for object: 0x7fff15ad6840 : ODemo::Engine::~~Engine()
    <--- member for object: 0x7fff15ad6840 : ODemo::Engine::~~Engine()
<--- function body: int app_example()
```

Possible Macro's

COMP MISC FUNCTION TRACER:

- Placed at the start of a function

COMP MISC BLOCK TRACER:

- Placed at the start of a compound statement (Block)

COMP MISC LOG TRACER:

- Provide a manual entry to the output (print statement)

COMP MISC MEMBER TRACER:

- Placed at the start of a member function

Important areas to practice

- Constructor types
- Assignment operators
- Inheritance
- Exception handling
- Conversions
- Member initialization
- Memory management
- Function and operator overloading

Using the object tracer

```
git clone https://github.com/thomasave/prog2.git
cd prog2
mkdir build
cd build
cmake ..
make install
./installed/bin/app_exec
cut -f 2 -d ] installedbinapp_exec.g3log.20201123-172103.log
```

Using the object tracer

Process the log with "cut -f 2 -d] <logfile>" to get a clean output.

The available tests are:

- app_copy
- app_essentialops
- app_example_exam_2021_1
- app_example_exam_2021_2
- app_exception
- app_fcalls
- app_hierarchies
- app_poly1
- app_poly2

Select a demo by name: app_copy

Exiting ...

g3log g3FileSink shutdown at: 17:21:12

Log file at: [./installedbinapp_exec.g3log.20201123-172103.log]

Using the object tracer

g3log created log at: Mon Nov 23 17:21:03 2020
message (uuu*: microsecond counter since initialization of log worker)

```
"Starting main program ..."  
"---> function body: int app_copy()  
"    statement: shared_ptr<Engine> e1Ptr(new Engine(100));"  
"    ---> member for object: 0x5614730ea090 : ODemo::Engine::Engine(double)"  
"    <--- member for object: 0x5614730ea090 : ODemo::Engine::Engine(double)"  
"    statement: Person* p1Ptr = new Person("Owner 1");"  
"    ---> member for object: 0x5614730e9b30 : ODemo::Person::Person(std::string)"  
"    <--- member for object: 0x5614730e9b30 : ODemo::Person::Person(std::string)"  
"    statement: Motorcycle* m1Ptr = new Motorcycle(e1Ptr, p1Ptr);"  
"    ---> member for object: 0x5614730ea840 : ODemo::Body::Body(std::string)"  
"    <--- member for object: 0x5614730ea840 : ODemo::Body::Body(std::string)"  
"    ---> member for object: 0x5614730e5f20 : ODemo::Wheel::Wheel()"   
"    <--- member for object: 0x5614730e5f20 : ODemo::Wheel::Wheel()"   
"    ---> member for object: 0x5614730e5f30 : ODemo::Wheel::Wheel()"   
"    <--- member for object: 0x5614730e5f30 : ODemo::Wheel::Wheel()"
```

(...)

Using the object tracer

- Example exercises are located at:

`prog2/main/cpp/objtracer/*`

- Practice using these provided examples!
- Edit these examples or follow [these instructions](#) to add your own!
- Don't underestimate these exercises, you'll be surprised by some of the outcomes

Using the object tracer

- You'll get the [doxygen documentation of prog2](#), including some new files
- We'll give you the code that needs to be traced
- Your task is to produce the expected log as if the prog2 object tracer was run

This Session

- Try to make the exercise from the January 2021 exam
(main/cpp/objtracer/example_exam/app_example_exam_2021_1.cpp)
- Verify your own solution using the object tracer
(choose app_example_exam_2021_1 demo)
- You likely won't finish it today, but hopefully understand how it works
 - Simply make sure to practise it enough for the exam

Questions

- Are there any questions?
- Send an e-mail to Thomas.Ave@uantwerpen.be