Advanced C++, Day 2

(C) Richèl Bilderbeek

March 23, 2015

Chapter 1

Today

1.1 Goal

- Compile-time speed
- Run-time speed
- Choosing the correct container
- Choosing the correct algorithm

1.2 Planning

- From NAG to C++
- Compile-time speed
- Run-time speed in general
- (Run-time speed and the STL)

Chapter 2

Go!

2.1 Warming up

- Lecture: From NAG to C++
- Lecture: Compile-time speed

2.2 Question

- What is the difference between compile-time and run-time polymorphism?

2.3 Answer

- Compile-time polymorphism: selecting a (member) function by using type selection
- Run-time polymorphism: selecting a member function by using inheritance

2.4 Question

```
//Never use switch,
//try a more scalable strategy instead
//Person must not have virtual member functions
//in the interface (Sutter, Exceptional C++, Item 23)
//Put every class in its own .h and .cpp file

int main()
{
  const Person happy(/* */);
  const Person grumpy(/* */);
  happy.SayHello(); //Should display 'Hi!'
  grumpy.SayHello(); //Should display '(mumble) hi'
  happy.SayBye(); //Should display 'Bye!'
  grumpy.SayBye(); //Should display '(mumble) bye'
}
```

2.5 Answer introduction

- The way to say hello must be a Strategy Design Pattern, e.g. SayHelloStrategy
- The way to bye hello must be a Strategy Design Pattern, e.g. SayByeStrategy
- A Person must have a Strategy to say hello
- A Person must have a Strategy to say bye
- Full answer at `www.richelbilderbeek.nl/CppStrategyDesignPatternExam` `htm`

2.6 Question

```
//Get the main to work (C++11: fix the compile error)
//Put every class in its own .h and .cpp file
//C++98: In strategy.h write 'enum Policy { A, B };'
//C++11: In strategy.h write 'enum class Policy { A, B };'

int main()
{
  const Strategy<A> x;
  const Strategy<B> y;
  x.DoIt(); //Displays 'Did it the A way'
  y.DoIt(); //Displays 'Did it the B way'
}
```

2.7 Answer

- strategy.cpp will be empty: cannot do forward declarations on templated member functions (yet?)
- See `www.richelbilderbeek.nl/CppCtStrategyDesignPatternExampleDoI`
  `htm`

2.8 Question

- What is the difference between compile-time and run-time polymorphism?
  - compile-time speed?
  - run-time speed?

2.9 Answer

- Compile-time polymorphism
  - Slower to compile
  - Faster to run: no virtual function calls
- Run-time polymorphism
  - Faster to compile
  - Slower to run: virtual function calls
- These two techniques are orthagonally related

2.10 Run-time speed

- Lecture: Improve run-time speed in general
- Lecture: STL and run-time speed

2.11 Exerise

- Every person has an GUID, a 32 digit number, which is created at random (and is therefore assumed to be unique)
- Task: GUID creation
- Every person has a name
- When a person log in, he/she receives a GUID
- Persons can be searched for from their GUIDs only
- The GUID and the person's info is removed when the person quits the current session
- Task: Management of persons
- Task: testing and profiling

Chapter 3

EOF