

# *Exercises week 1*

Klaas Isaac Bijlsma  
s2394480

David Vroom  
s2309939

November 18, 2017

## **Exercise 1**

*Attain some familiarity with the way functions are selected from namespaces*

We used the following code,

main.cc

```
1 #include <iostream>
2
3 namespace First
4 {
5     enum Enum
6     {};
7
8     void fun(First::Enum symbol)
9     {
10         std::cout << "First::fun called\n";
11     }
12 }
13
14 namespace Second
15 {
16     void fun(First::Enum symbol)
17     {
18         std::cout << "Second::fun called\n";
19     }
20 }
```

```

21 |
22 | int main()
23 | {
24 |     First::Enum symbol;
25 |
26 |     fun(symbol);           // First::fun called
27 | }

```

**Call fun and explain why First::fun is called. How would you call Second::fun instead?**

Als een functie uit een namespace wordt aangeroepen zonder de namespace te specificeren, dan wordt de namespace van het argument van de functie gebruikt om de namespace van de functie te bepalen; het zogenaamde 'Koenig Lookup'. Aangezien het argument is gedeclareerd als type `First::Enum` wordt `First::fun` aangeroepen. Om `Second::fun` aan te roepen moet de namespace expliciet worden genoemd: `Second::fun(symbol)`.

**In the namespaces slides (#6) it is stated that operator<<'s use is simplified because of the Koenig lookup. Explain.**

Zonder Koenig lookup zal de korte versie `std::cout << "Hello"` (net als `operator<<(std::cout, "Hello")`) niet gebruikt kunnen worden. De insertion operator functie uit de standard namespace zou dan niet bereikbaar zijn zonder de prefix `std::` en de expliciete functie call `operator<<(std::cout, "Hello")`.

**Now, just above main, declare a function void fun(First::Enum symbol). Compile this program. What happens? Why?**

Er ontstaat een ambiguititeit. De compiler weet nu niet of hij de functie uit de namespace `First` of de globale functie net boven main moet aanroepen.

## Exercise 2

*ziet ie dit?*

## Exercise 3

## Exercise 4

## Exercise 5

## Exercise 6

## Exercise 7



## Exercise 8

## Exercise 9

## Exercise 10