

# Notities les 5

Jelle De Bock

November 3, 2015

## 1 Recap vorige les c++

Referentietype: *[const] type &var*

eigenlijk wordt het adres meegegeven, is perfect om te gebruiken bij methodes in C++. Ook zeker const gebruiken wanneer het kan.

Console invoer en uitvoer: *cin en cout*. Je kan ze concateneren.

```
cout << "Het_getal_is" << getal; //uitschrijven
cin >> naam;
getline(cin, naam);
```

Getline na cin kuist de lijn van cin op (de enter). Get() vs getchar() get aanvaardt ook whitespaces...

## 2 Namespace

Erg vergelijkbaar met packages in Java. Dienen om collisions tussen functies met dezelfde naam uit verschillende libs te vermijden. Verschillende manieren van oproeping:

```
using std::string;
using std::cout;
using namespace std;
```

Opgelet wanneer er nog eens een functie met dezelfde naam in uw file staat. Dan kan de compiler niet weten de welke je bedoelt (die uit de package of die in de file).

## 3 ios package

Een package die schrijven van / naar files mogelijk maakt.

### 3.1 Oefeningetje

De gebruiker mag 5 lijnen tekst ingeven. Deze woorden toegevoegd achteraan een bestand.

```

#include <fstream>
using namespace std;

int main()
{
    string lijn;
    ofstream bestand("tekst.txt", ios::app);

    for(int i=0; i<5; i++)
    {
        getline(cin, lijn);
        bestand << lijn << endl;
    }
    bestand.close();
}

```

### 3.2 slide 37

Wat doe het programmafragment?

Bestand openen voor lezen en schrijven.

Lees getal in tot er geen meer staat,

schrijf het gewijzigde getal weg. MAAR: schrijven en lezen tegelijkertijd is dikke miserie...

### 3.3 Opmerkingen

Telkens je een ifstream wil gebruiken moet je de referentieoperator gebruiken.

## 4 Malloc en calloc varianten

Je mag *malloc* en *calloc* en de daarbijhorende *free* nog gebruiken, maar je kan nu ook de modernere *new* en *delete* gebruiken. Voor arrays gebruik je *new[]* en *delete[]*