## Installeer

### Installeer en configureer (voor zover je dit nog niet hebt gedaan voor vorige cursussen) Python, CodeLite, GIT, MinGW, bmptk, en de voorbeeld code.

De installatie is voltooit en werkt naar behoren.

## Boeken

### Wat doet de contructor van de klasse book?

De constructor van de klasse book zorgt ervoor dat tekst, author en title goed worden gedefineerd. In dit geval heeft de programmeur een eigen constructor gemaakt die de paramethers tekst, author en title neemt. Deze paramethers worden gebruikt op de variabele van de aangemaakte object te definieren die dezelfde naam hebben. Er wordt hier gebruik gemaakt van een initialisatie lijst in plaats van de variabele in de body goed te zetten.

### Hoe noemen we de tekst tussen de : (dubbelepunt) en de { (krulhaak) bij de constructor?

Dit wordt de initialisatie lijst genoemd. Deze initialisatie lijst is erg handig als je gebruikt maakt van variables by reference, omdat je dan geen foutmeldingen krijgt. Een object moet namelijk van deze variabele van tevoren weten waar het naartoe wijst anders loopt je programma in de soep als je functies aanroept die van deze variable by reference gebruik maakt.

### Welke functies van de klasse book zijn van buitenaf toegankelijk?

In dit geval is de constructor van buitenaf toegankelijk, maar je kan erover discusseren of dit daadwerkelijk een functie/method is. Daarnaast is print toegankelijk van buitenaf. De overige functies zijn private. Dus niet benaderbaar van buitenaf.

### Waarom denk je de schrijver van deze code er voor heeft gekozen sommige functies private te maken?

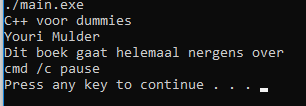
Het is in sommige gevallen niet verstandig om bepaalde functies/methods van buitenaf benaderbaar te maken, omdat er dan dingen aangepast kunnen worden door andere functies die niet horen te kunnen. In dit geval gaat het om print\_text(), print\_author() en print\_title(). Deze functies/methods heeft de programmeur niet nodig buiten de class, omdat print() ze allemaal tegelijk al aanroept.

### Waarom staat er in de .cpp voor elke functie “book::”?

Als de programmeur dit niet doet weet de compiler niet bij welke klasse dit hoort, omdat deze methods buiten de klasse hun definitie. Als je ze definieert in de klasse is de book:: niet nodig. Als book:: er niet voor staat krijg je een error, omdat de compiler dan niet weet bij welke klasse deze functie hoort.

## Main

### Maak een project aan met de book.cpp en book.hpp, en een main.cpp waarin de main functie een book aanmaakt en deze print.



## Meubels

### Maak een project aan met de klassen: *stool*, *table* en *furniture*. *Stool* heeft als member variabelen n\_legs en n\_seats. *Table* heeft als member variabelen n\_legs, length en width. De constructors krijgen de aantallen mee en slaan die op. De membervariabelen zijn allemaal private.

### *Furniture* bestaat uit 4 variabelen van het type *stool* en 1 variabele van het type *table*. De constructor van *Furniture* initializeert de meubelen met een redelijke waarden. De membervariabelen van *Furiniture* zijn ook private.

### *Furniture* heeft de methode *makeMoreHipster*(). Deze functie haalt van alle 4 de stoelen één poot af en geeft de tafel deze 4 poten. Schrijf deze functie en de benodigde functie’s bij *stool* en *table.*

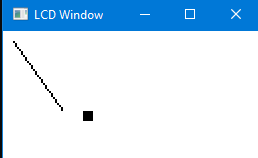
### Schrijf de main functie. Deze maakt een *Furniture* aan, print zijn eigenschappen, roept zijn *makeMoreHipster*() aan, en print weer de eigenschappen. Om dit te kunnen doen moet je print functies toevoegen aan alle klassen.

### Teken een UML-diagram met *stool*, *table* en *furniture*.

## Gevulde rechthoek

|  |
| --- |
| D:\2017-2018\vakken\V1OOPC\images\h1\filled-rectangle.png |
| UML diagram van een opgevulde rechthoek (en de window waarop hij wordt afgebeeld) |

### Neem de 04-09-window voorbeeldcode als uitgangspunt. Schrijf (voeg er aan toe) een klasse (hpp en cpp files) voor een opgevulde rechthoek. De constructor ziet er hetzelfde uit als van de rechthoek in de voorbeelden maar bij het afdrukken moeten alle pixels die deel uitmaken van de rechthoek (inclusief de randen) getekend worden. Je maakt dus geen gebruik van de lijn klasse, gebruik direct de methode(n) van window. Demonstreer je klasse in een applicatie die een aantal gevulde rechthoeken toont. In het UML diagram zijn de attributen van de gevulde rechthoek weggelaten: die moet je zelf bedenken.



## Samengesteld object

### Gebruik de code van de vorige opgave als uitgangspunt. Maak een klasse voor een grafisch object (cpp en hpp files) dat (ten minste)

### 1 rechthoek (de bestaande of de in de vorige opgave gemaakte gevulde rechthoek),

### 1 lijn en

### 1 circkel bevat (dus een compositie)

### en van ten minste 1 van deze basis objecten meer dan 1.

### Verdere vereisten:

### Het grafische object moet iets herkenbaars zijn, bv. een huis, trein, wolkenlucht, fabriek, pacman.

### De constructor van jouw object moet (ten minste) 1 parameter hebben die iets aan het object aanpast, bv voor een trein het aantal wagons.

### Maak ook een UML diagram met jouw klasse, de basis klassen die jouw object gebruikt en de window klasse.

### Demonstreer jouw klasse in een applicatie die er (ten minste) twee verschillende varianten van afbeeldt.