# Toets: Pizza tijd

Als er veel toetsen nagekeken moeten worden, kan het wel eens zo zijn dat de docenten geen tijd meer hebben om een fatsoenlijke maaltijd klaar te maken. Gelukkig is er dan nog altijd de mogelijkheid om pizza te bestellen om snel wat te eten te krijgen. Het pizza restaurant heeft door gekregen dat het om docenten van de afdeling HBO-ICT gaat en heeft gevraagd om of wij kunnen helpen om digitaal bonnetjes te maken. Het is nu jouw taak om een systeem te ontwikkelen die de bonnetjes van een pizza bestelling kan uitprinten.

De bestellingen die gedaan kunnen worden bij het restaurant zijn gelukkig eenvoudig. Er zijn twee soorten producten: namelijk frisdrank en pizza’s. Voor frisdrank moet de prijs per blikje en het aantal worden ingevuld om de totaal prijs te kunnen berekenen. Voor pizza’s is de prijs wat lastiger te berekenen, omdat er veel verschillende pizza’s zijn. In overleg met het restaurant is besloten dat de werknemer zelf de lijst van ingrediënten en de prijs per ingrediënt invoert. Uit de lijst van ingrediënten kan dan de totaalprijs van de pizza uit worden berekend.

Voor het systeem hoeft geen database van mogelijke frisdranken en pizza’s bij te houden, ook hoeft geen database van mogelijke ingrediënten worden bijgehouden. De werknemer vult per pizza in welke ingrediënten op de pizza komen. Het printen van de bestelling en de totaal prijs gebeurt door het schrijven naar std::cout.

**Let op:** Je hoeft geen menu of andere userinterface te maken voor het systeem. Het volstaat het om een main functie te maken waarin je een aantal frisdranken en pizza’s toevoegt (zie het bijgeleverde main.cpp bestand voor details).

## Voorbeelduitvoer

Uitvoer van het systeem zou er als volgt uit kunnen zien:

Welcome to Cheesy Pizzas, you have ordered the following products:

3x Soft-Drink Coca-Cola, price: 2.15

1x Soft-Drink Orange-Juice, price: 2.65

2x Pizza Margherita, price: 6.75

Ingredients: Dough (3) Tomato (2) Mozzarella (1) Oregano (0.75)

2x Pizza Salami, price: 7.75

Ingredients: Dough (3) Tomato (2) Paprika (1) Salami (1.75)

Total price of your order: 38.1

**Let op:** Lees ook de achterkant voor tips!

## De stappen om het systeem te bouwen

* Ontwerp het systeem op kladpapier. Bedenk alle classes en hun attributen en methodes. Teken de meest optimale oplossing, je hoeft het ontwerp niet in te leveren.
* Lees main.cpp aandachtig door en kijk wat je systeem uiteindelijk moeten kunnen doen. Eén van de onderdelen is het inlezen van een bestand. Dit bestand is bijgeleverd en mag niet aangepast worden. Je kunt het bestand openen als “input.txt”.
* Schrijf de benodigde classes. Zorg voor goede output() methodes voor elke class, deze methode schrijft de class naar ‘std::cout’.
* Schrijf de Order class die de bestelling bevat.
* **Let op:** Zorg ervoor dat aan het einde van het programma ook alle classes zijn opgeruimd en alle geheugen is vrijgegeven. Aan het eind van de applicatie mag je dus geen geheugen laten leaken!
* **Let op:** Iedere class **moet** gedeclareerd worden in zijn eigen header file en iedere implementatie van de class in zijn eigen cpp file. Je mag dus niet alle code in de main.cpp zetten, je moet de classes opsplitsen over verschillende source en header files.

## Enkele tips

Omdat we begrijpen dat best lastig kan zijn om een dergelijk systeem te bouwen hebben we hier enkele tips:

* Test tijdens het bouwen voortdurend. Dit kun je doen door aanroepen vanuit je *main* te doen naar je Receipt class en eventueel wat te printen. Zorg wel dat je aan het einde alleen alle stappen doet die aangegeven zijn in het commentaar in de main, zodat je onderstaande uitvoer krijgt.
* Zijn er delen waar je niet uit komt? Probeer er dan omheen te werken zodat je toch nog punten kunt scoren op de andere onderdelen. Beschrijf eventueel in commentaar regels wat je probeert te doen, zodat de beoordelaar snapt waarom je bepaalde delen wel of niet gedaan hebt.
* Maak gebruik van het feit dat je de website cplusplus.com en cppreference.com kunt benaderen!  
   **Let op: De zoekfunctie op cplusplus.com werkt niet omdat google.com is geblokkeerd en deze zoekfunctie van google.com gebruik maakt.**
* Voor de duidelijkheid. We beoordelen op:
  + De werking van je programma en de netheid van je code
  + Het verdelen van de verantwoordelijkheden over de classes, je individuele classes en het gebruik van overerving en access-specifiers
  + Het correct gebruik van header en cpp files en het uitsplitsen van classes over verschillende source file
  + Het gebruik van STL datastructuren en algoritmen
  + Correcte allocatie en deallocatie van geheugen
  + Correct gebruik van method/functie argumenten (by-value-, by-reference, by-address) en gebruik van const methoden waar mogelijk