

## Praktijktoets CMTTHE01-4 (onderdeel programmeren 4)

Heb je een **even** studentnummer dan werk je **opdracht A** uit

Heb je een **oneven** studentnummer dan werk je **opdracht B** uit

Bij de start van de toets (zie rooster jaar 1) zullen de opdrachten toegankelijk worden via [Cumlaude](#).

De opdracht bevat een beschrijving van een spelletje. Vervolgens heb je 4 uur de tijd om de objectgeoriënteerde code te ontwerpen en uit te werken. Binnen de deadline van 4 uur lever je een zip-bestand in op Cumlaude. Hiervoor staat een inleveropdracht klaar.

De zip die je download bevat startcode en een bestand “verantwoordingsdocument.txt”, schrijf hierin je verantwoording.

De zip die jij upload bevat de volgende onderdelen:

- Verantwoordingsdocument met verantwoording van de keuzes uit het klassendiagram en de code
- Klassendiagram
- Code

Plaats het verantwoordingsdocument in de map van het codeproject.

Het klassendiagram werk je uit op papier (liefst met pen). Hiervan maak je een foto (of een scan) en deze plaats je ook in de root folder van het codeproject.

Van de root folder maak je een zip-bestand. Verwerk je studentnummer en naam in de file name.

De inleveropdracht sluit automatisch na 4 uur. Wanneer er binnen de deadline geen zip-bestand is ingeleverd, volgt een “ND” als beoordeling in Osiris.

Omschrijving	Becoördeling	Deliverables
<b>Classes en verantwoordelijkheden</b>		
De spelregels van het spel en de bijbehorende verantwoordelijkheden moeten belegd zijn in de classes in het klassendiagram. In het verantwoordingsdocument is verantwoord waarom de classes die verantwoording hebben	Elke verantwoordelijkheid is belegd 1 punt 1 verantwoording mist, 0 punten	Verantwoordingsdocument
	De juiste classes zijn benoemd in het klassendiagram, 1 punt 1 class mist of wijkt af, 0 punten	Klassendiagram
Het is minimaal 1x gelukt om een class toe te passen in code en hiervan is met succes een object aangemaakt.	Goed toegepast 1 punt Niet goed toegepast 0 punten	Code
<b>Encapsulation</b>		
In het klassendiagram is op de juiste manier aangegeven wat de <i>access modifier</i> (+, -, #) moet worden van alle members In het verantwoordingsdocument is van alle afwijkende members (public en protected members) een verantwoording gegeven.	Je kunt uitleggen waarom je de access modifiers op die plek gebruikt, 1 punt Je kunt niet uitleggen waarom je de access modifiers op die plek gebruikt, 0 punt	Verantwoordingsdocument
	De access modifiers zijn nuttig voor deze members op de gebruikte plek, 1 punt 1 access modifier is niet nuttig voor een member, 0 punten	Klassendiagram
Het is minimaal 1x gelukt om een getter en/of setter toe te passen in code en er in code ook gebruik van te maken	Goed toegepast 1 punt Niet goed toegepast 0 punten	Code
<b>Composition</b>		
In het klassendiagram is op de juiste manier een composition aangegeven en dit is op een toepasselijke plek. Ook is hier de juiste notatie voor gebruikt.  In het verantwoordingsdocument is verantwoord waarom de composition op die plek is toegepast.	Je kunt uitleggen waarom je composition op een bepaalde plek gebruikt hebt, 1 punt Je kunt niet uitleggen waarom je composition op een bepaalde plek gebruikt hebt, 0 punten	Verantwoordingsdocument
	Composition is op de juiste manier getekend, 1 punt Composition is niet op de juiste manier getekend, 0 punten	Klassendiagram
Het is minimaal 1x gelukt om een functie van een compositie te gebruiken.	Goed toegepast 1 punt Niet goed toegepast 0 punten	Code
<b>Inheritance</b>		
In het klassendiagram is minimaal 1 x op een juiste wijze overerving toegepast en dit was nuttig op die plek  In het verantwoordingsdocument is verantwoord waarom op die plek overerving is toegepast	Je kunt uitleggen waarom je overerving op een bepaalde plek gebruikt hebt, 1 punt Je kunt niet uitleggen waarom je overerving op een bepaalde plek gebruikt hebt, 0 punt	Verantwoordingsdocument
	Inheritance is op de juiste manier getekend, 1 punt Inheritance is niet op de juiste manier getekend, 0 punten	Klassendiagram
Het is minimaal 1x gelukt inheritance toe te passen.	Goed toegepast 1 punt Niet goed toegepast 0 punten	Code

## Toetsmatrijs

TAXONOMIE	LEERDOELEN	TOETSING
Fundamental Knowledge	<p>Ik kan de betekenis en werking van de volgende OOP principes herkennen en uitleggen</p> <ul style="list-style-type: none"><li>• Classes</li><li>• Inheritance</li><li>• Composition</li><li>• Encapsulation</li></ul>	<p>Klassendiagram (herkennen) Verantwoordingsdocument (uitleggen)</p>
Application	<p>Ik kan de betekenis en werking van de volgende OOP principes zelf implementeren</p> <ul style="list-style-type: none"><li>• Classes</li><li>• Inheritance</li><li>• Composition</li><li>• Encapsulation</li></ul>	<p>Praktijkopdracht Toepassen van de code</p>
Fundamental Knowledge	<p>Ik kan code van anderen lezen en de werking ervan beredeneren.</p>	<p>Praktijkopdracht Doorgronden van aangeleverde startcode</p>
Fundamental Knowledge	<p>Ik kan een programmeervraagstuk op abstract niveau structureren en oplossen middels een klassendiagram (UML) waarbij gebruik gemaakt wordt van Inheritance, Composition en Encapsulation.</p>	<p>Klassendiagram</p>