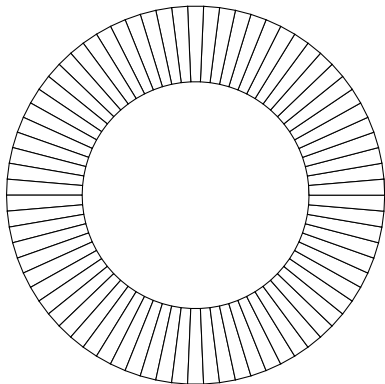


ALGORITMISCH DENKEN

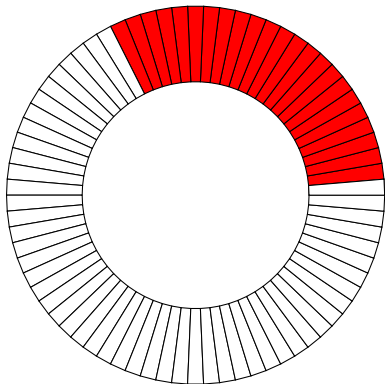
INTRODUCTIE

Studiebelasting



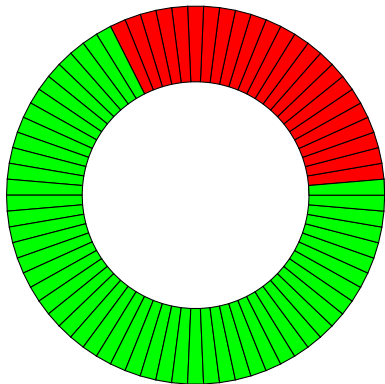
- 3 studiepunten
- 25 uur per studiepunt
- 12 weken
- \Rightarrow 75 uren in totaal

Studiebelasting



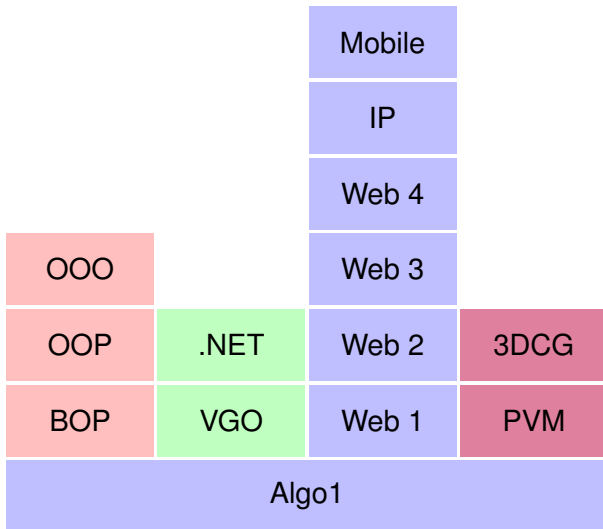
- 3 studiepunten
- 25 uur per studiepunt
- 12 weken
- \Rightarrow 75 uren in totaal
- 2 uur/week theorie of oefeningen

Studiebelasting



- 3 studiepunten
- 25 uur per studiepunt
- 12 weken
- \Rightarrow 75 uren in totaal
- 2 uur/week theorie of oefeningen
- 50 uur zelfstudie

Plaatsing (Onvolledig Overzicht)



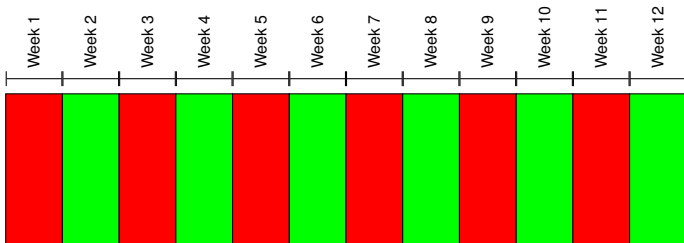
Algemeen

- Leermateriaal
 - Zie Toledo
 - <http://algo.leone.ucll.be>
- Evaluatie
 - 100% punten op examen
- Officiële informatie: zie studiewijzer

Inrichting Vak

- Nadruk ligt op kunnen, niet kennen
- Werken voor dit vak = oefeningen maken
- Examen bestaat ook uitsluitend uit oefeningen

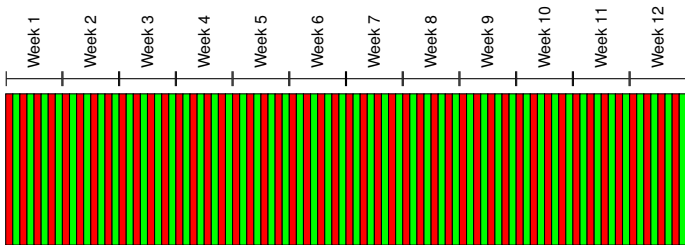
Inrichting Vak



Officieel Uurrooster

- Weken afwisselend 2 uur les en labo
- Veel te rigide
 - Twee uur stilzitten en luisteren is te veel gevraagd
 - Te veel tijd tussen theorie en oefening
 - Er is onvoldoende theorie om 12 uur vol te praten

Inrichting Vak



Wat wij gaan doen

- Mondelinge uitleg zo kort mogelijk houden
- Snelle afwisseling tussen theorie en oefening
- Nadruk op praktijk

Inrichting Vak

- Niveauverschil programmeren tussen studenten enorm
 - Sommigen geen programmeerervaring
 - Sommigen geleerd in middelbaar
 - Sommigen komen van de KUL
- Klassikaal uitleg geven problematisch
 - Voor de ene helft gaat het te snel
 - Voor de andere helft gaat het te traag
 - → Aandacht van ganse groep is verloren gegaan
- Oplossing
 - Alle uitleg staat in geschreven vorm
 - Iedereen werkt tegen eigen tempo
 - In het begin van sessie korte uitleg

Hoe Aanpakken?

- Oefeningen zijn eerder raadsels/puzzels
- Eenmaal je de oplossing kent, heeft het geen zin een raadsel opnieuw op te lossen
- Dezelfde oefeningen herhalen heeft dus weinig nut
- → Groot aantal oefeningen beschikbaar

Code Schrijven

- Code wordt niet “lineair” geschreven
- Code wordt in kleine stukjes opgebouwd
 - Aanpassen
 - Invoegen
 - Verwijderen
 - Verplaatsen
- Men probeert na elke wijziging te testen
- Te veel studenten schrijven code in grote brokken
 - Weten niet of wat ze tot nog toe geschreven hebben werkt
 - Als er een fout blijkt te zijn, is die moeilijker te vinden