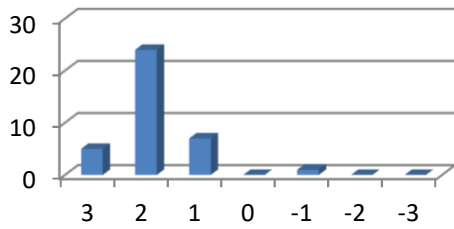
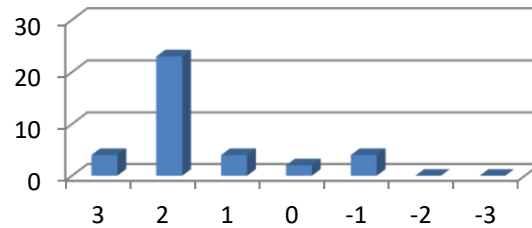


Enquete cursus informatica 1e bachelors 2016-2017

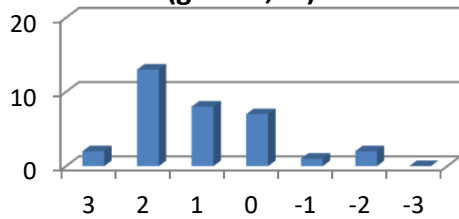
Python (gem=1,86)



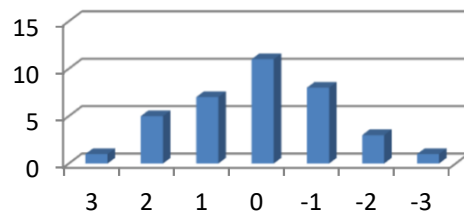
Java (gem=1,57)



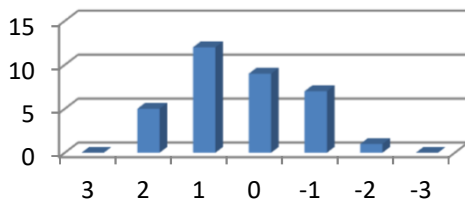
Combinatie python – java (gem=1,06)



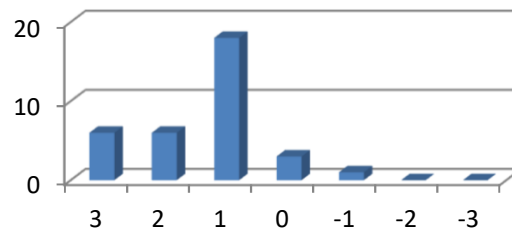
Hoe is de overgang python2java



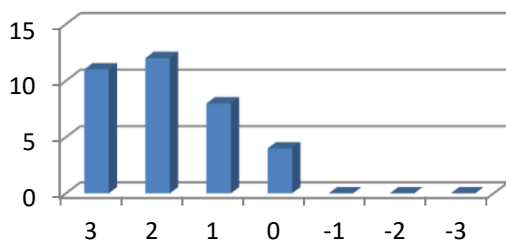
Java-oefeningen (gem=0,38)



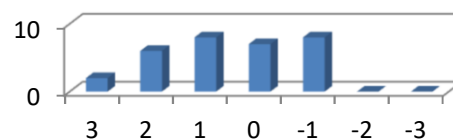
Deel I nuttig voor begrijpen java & OO



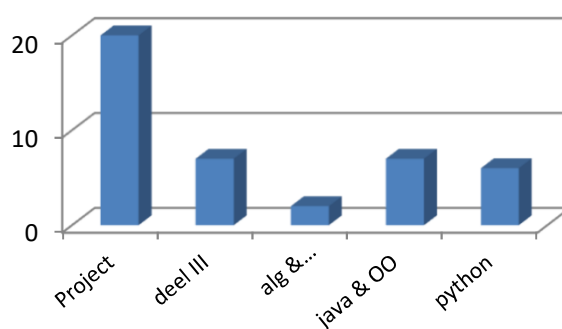
Java project (gem=1,86)



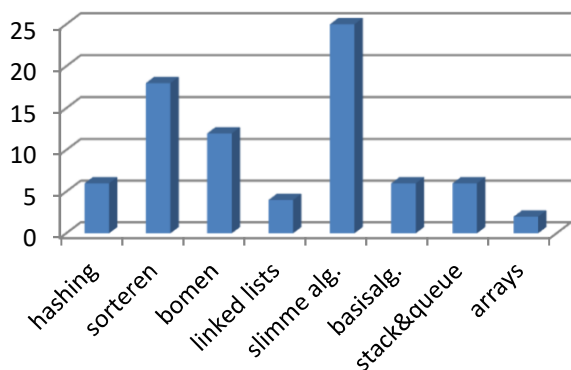
oefeningen beter aansluiten bij project? (gem=0,58)



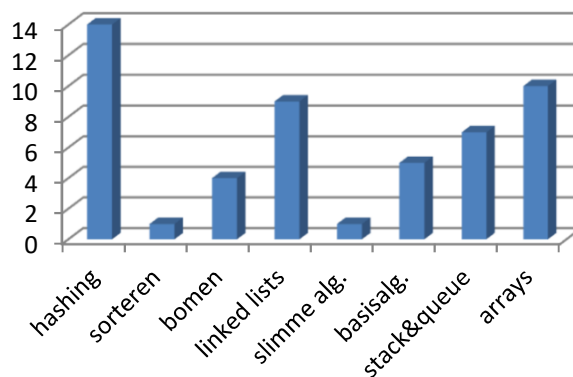
Tofste deel van cursus



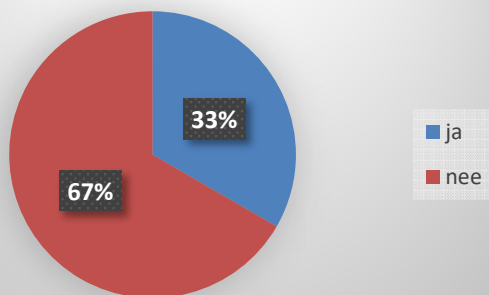
Deel II meest interessante hfst



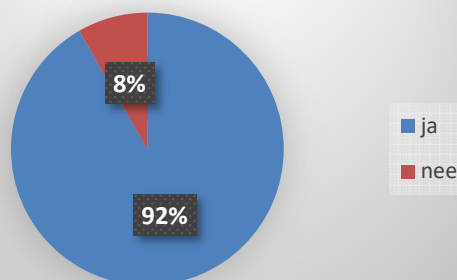
Deel II minst interessante hfst



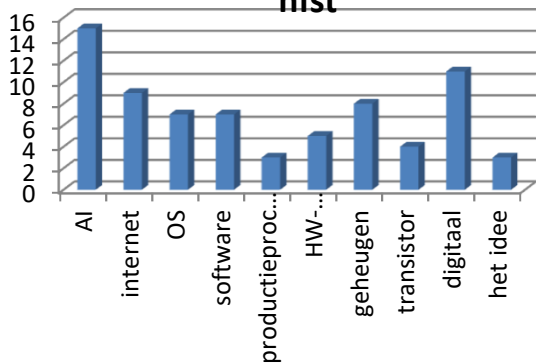
Online code gebruikt?



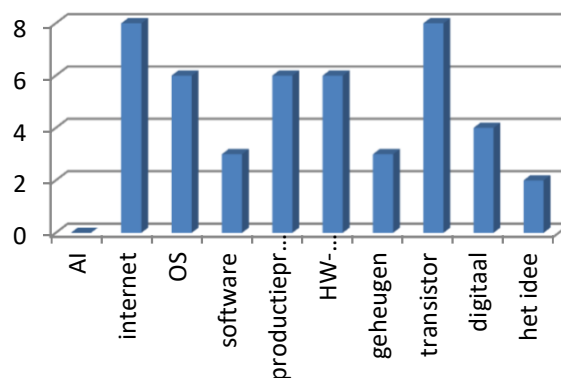
Online code nuttig?



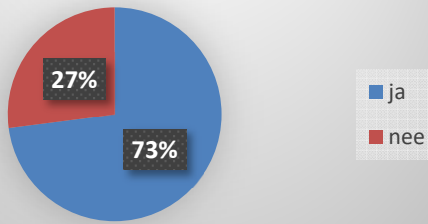
Deel III meest interessante hfst



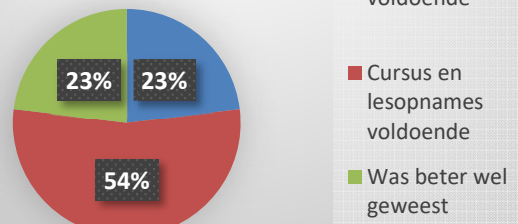
Deel III minst interessante hfst



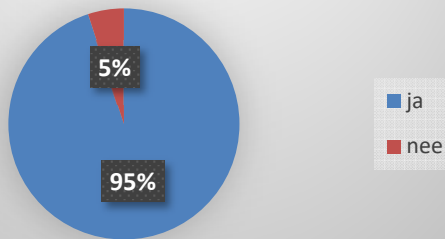
Ben naar de les geweest



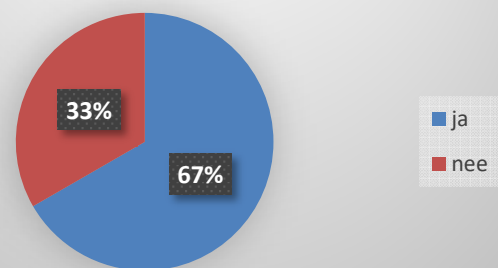
Ben niet naar de les geweest



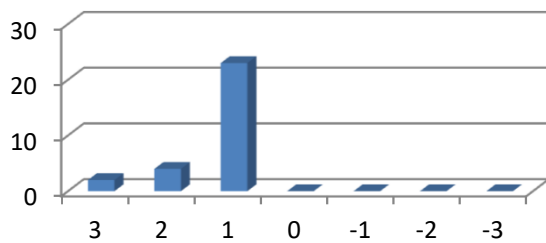
Ben naar de les geweest, vond HOC nuttig



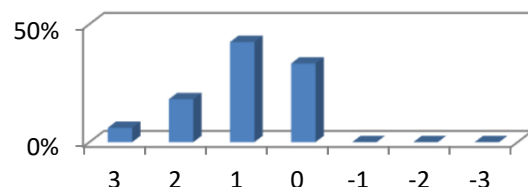
Ben naar de les geweest, heb lesopnames bekeken



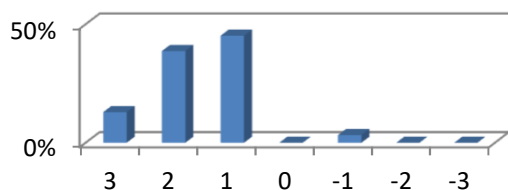
stukjes historiek interessant



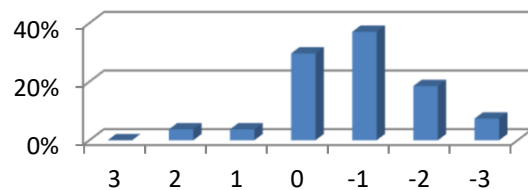
Geïnteresseerd in extra HOC met advanced topics?



Is het getoonde onderzoek interessant



Kortere HOCs, details aan zelfstudie overlaten?



commentaren van enquete cursus informatica 1e bachelors 2016-2017

python-theorie

python-oefeningen

meer belichten: recursie, tekstverwerking, strings (kwam op examen voor!)

mogen iets complexer zijn, als voorbereiding op java

meerdere oefeningen van dezelfde soort

java-oefeningen

meer belichten: GUI (3), JFrame & JLabel, abstracte klassen/interfaces (2), abstractie, overerving, strings, lijsten

meer op basis focussen (2)

meer oefeningen

te snel (doorgerushed) - te weinig tijd om alles te begrijpen (2)

examenwaardige oefeningen (2)

de WPOs waren niet echt bruikbaar in het begin, want java is vrij abstract, dus we wisten vaak niet wat de bedoeling was. Pas bij project begin je alles te begrijpen.

project

meer aandacht: hoe herhaalde code in superklasse zetten, hoe er aan beginnen (2), globaal concept geven, debuggen, meerdere klassen gebruiken, switchen tussen panels, aanmaak van levels, timergebruik (2), met pixels werken, gebruik van juiste datastructuren, overerving, abstracte klassen

suggestie: GUI via template en meer aandacht geven aan algoritmes

we hebben java voor specifiek gezien, moeten meer algemeen bekijken

Deel I java

ontbreekt/kan beter: gebruik van chars, abstractie, interfaces, manipulatie van strings, matrices, methodes, overbrengen van info van 1 klasse naar een andere

HOC

meer chronologisch werken

kwaliteit lesopnames kan beter (bord is niet te lezen) (2)

theorie

positief: opnemen van ideeën van studenten + optionele informatie voor goede programmeurs

beter uitleggen (Deel II): astar, hashing (2), minimax

beter uitleggen (Deel III): OS (2), internet (2)

Deel III: beter een cursus

iets minder economie op het einde

algemeen

les over hoe je code van groot project kan begrijpen

Enquete over inhoud cursus

Vrijblijvend mag je feedback geven over de *inhoud*. (De kwaliteit evalueer je via de officiële beoordelingsprocedure van de VUB.) Hoe beter je aangeeft wat & hoe 't beter kan, hoe bruikbaarder je feedback. Vul je commentaar/suggesties in en/of

-3	zeer negatief	-2	negatief	-1	eerder negatief
		0	neutraal		
+1	eerder positief	+2	positief	+3	zeer positief

(a) Hoe evalueer je de combinatie python – java? ...

Python: aanleren basis van programmeren (for, while, array, functie, recursie) ...

Java: object-georiënteerd programmeren ...

Hoe is de overgang?

(b) Welk vond je het tofste/interessantste deel

Python - Java & Object-oriëntatie - Algoritmen & Datastructuren (deel II) – Deel III - Project

(c) Pythonoefeningen ...

Zouden meer de volgende onderwerpen moeten belichten: ...

Java-oefeningen ...

Zouden meer de volgende onderwerpen moeten belichten: ...

(d) Project ...

De java-oefeningen zouden beter mogen aansluiten met het project (of is dit niet nodig)? ...

Welke onderwerpen waren moeilijk/vergen meer aandacht:

(e) Deel I.: Was dit nuttig om Java & Object-oriëntatie te begrijpen?

Welke onderwerpen ontbraken of mochten in meer detail:

(f) Deel II: algoritmen & datastructuren

Hoofdstuk 2: arrays – 3:stack&queue – 4:basialgoritmen – 5: slimme algoritmen – 6:gelinkte lijsten – 7:bomen – 8:sorteren – 9: hashing

- Welk(e) hoofdstuk(ken) vond je het meest interessant? 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9
- Welk(e) hoofdstuk(ken) vond je het minst interessant? 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

- De code van de cursus staat online

Heb je ze gebruikt? Ja / nee Indien ja, vond je ze nuttig: Ja / Nee

(g) Deel III: technologie, historiek en economische aspecten

Hoofdstuk 0: idee – 1: digitaal – 2: transistor – 3: geheugen – 4: architectuur – 5: productieproces – 6: software – 7: operating system – 8: internet – 9: artificiële intelligentie

- Welk(e) hoofdstuk(ken) vond je het meest interessant? 0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9
- Welk(e) hoofdstuk(ken) vond je het minst interessant? 0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9
- Vind je de (niet-te-kennen) stukjes historiek interessant?

(h) Hoorcolleges & lesopnames:

- Ik heb meestal wel aan de hoorcolleges deelgenomen
 - Vond ze nuttig om de leerstof te begrijpen:
 - Heb toch ook de lesopnames (gedeeltelijk) bekeken bij het blokken:
- ik ben meestal niet naar de hoorcolleges gegaan
 - was niet nodig, de cursus was voldoende:
 - was niet nodig, de lesopnames + cursus waren voldoende:
 - was beter wel naar de les gegaan:

(i) Varia

- Vind je het onderzoek dat ik toon interessant of onnodig/tijdverlies?
- Wat als ik niet alles vertel, details (bvb implementaties) overlaat aan zelfstudie (met video-opnames) en zo de hoorcolleges korter hou:
- Zou je geïnteresseerd zijn in een advanced HOC met advanced topics (die niet te kennen zijn):

Welke dingen waren nog onduidelijk van de cursus?

Verdere commentaar/feedback/suggesties:

Hartelijk Dank!!

Het Python & Java-team