

Leerdoel: De student kan...	Gewicht (%)	Uitstekend (100%)	Goed (75%)	Voldoende (50%)	Onvoldoende (0%)
een vooraf gegeven taak vertalen in een ontwerp (bijv. Nassi-Shneidermandiagram) van een programma (algoritme), en hierbij correct gebruik maken van selection, iteration en functies.	10	Er zijn PSD's of andersoortige diagrammen opgenomen om de applicatieworkflow te beschrijven. Deze zijn correct en helder beschreven. Er zijn extra functies opgenomen in het ontwerp.	Er zijn PSD's of andersoortige diagrammen opgenomen om de applicatieworkflow te beschrijven. Deze zijn correct en helder beschreven. Er zijn geen extra functies opgenomen in het ontwerp.	Er is voldaan aan de opdracht. Alle deelproducten zijn aanwezig. Het ontwerp is duidelijk, maar er is geen of weinig beschrijving van het ontwerp.	Er is niet (volledig) voldaan aan de opdracht, of er zijn geen ontwerpen aanwezig.
een ontworpen programmastructuur vertalen in Python-code (implementatie), en daarbij keuzes in de syntax kunnen onderbouwen.	5	De code komt overeen met de ontwerpen. Mocht dit niet het geval zijn, dan zijn verschillen gedocumenteerd. Keuzes in het ontwerp zijn onderbouwd. Er zijn geen overbodige bestanden of klassen, en functies en routines worden hergebruikt waar dat kan.	De code komt overeen met de ontwerpen. Mocht dit niet het geval zijn, dan zijn verschillen gedocumenteerd. Keuzes in het ontwerp zijn niet of slecht onderbouwd. Er zijn geen overbodige bestanden of klassen, en functies en routines worden hergebruikt waar dat kan.	De code komt in enige mate overeen met de ontwerpen. Verschillen zijn niet gedocumenteerd en keuzes zijn niet onderbouwd.	De code komt niet overeen met de ontwerpen.
verschillende alternatieven / algoritmen genereren voor een gegeven taak (divergeren in flows).	5	Studenten laten zien dat er meerdere oplossingen zijn voor hetzelfde probleem (bijv. for/while-loop of algoritme), door bijvoorbeeld verschillende (nieuwere) versies te laten zien waarbij ze code efficiënter maken. Ze houden hierbij rekening met de leesbaarheid (readability) en overdraagbaarheid (reusability) van de code. Naast deze maatregelen, wordt er ook gekeken naar performance, zo wordt er bijvoorbeeld ".join(seq)" gebruikt voor string-concatenatie, of wordt er gemeten hoeveel tijd/geheugen het programma in beslag neemt.	Studenten laten zien dat er meerdere oplossingen zijn voor hetzelfde probleem (bijv. for/while-loop of algoritme), door bijvoorbeeld verschillende (nieuwere) versies te laten zien waarbij ze code efficiënter maken. Ze houden hierbij rekening met de leesbaarheid (readability) en overdraagbaarheid (reusability) van de code.	Studenten laten zien dat er meerdere oplossingen zijn voor hetzelfde probleem, door bijvoorbeeld verschillende (nieuwere) versies te laten zien waarbij ze code efficiënter maken.	Studenten laten niet zien te beschikken over verschillende alternatieven voor bijvoorbeeld een for-loop (while-statement).
basic input- en output-functies (I/O) toepassen in een script (screen, keyboard en files)	10	Er is gebruik gemaakt van een database om gegevens op te slaan, bijvoorbeeld sqlite. Gegevens worden één keer ingelezen als dat maar één keer hoeft. Er is rekening gehouden met mogelijke problemen rondom de input, bijv. een nummer waar een string wordt verwacht.	Er is gebruik gemaakt van een database om gegevens op te slaan, bijvoorbeeld sqlite. Gegevens worden één keer ingelezen als dat maar één keer hoeft. Er is rekening gehouden met mogelijke problemen rondom de input, bijv. een nummer waar een string wordt verwacht.	Er is gebruik gemaakt van een simpel bestand om gegevens op te slaan, bijvoorbeeld in .txt of .csv formaat. Bestanden worden meerdere keren ingelezen, terwijl dat maar één keer nodig is. Er is geen rekening gehouden met mogelijke problemen rondom de input.	De gegevens zijn niet correct opgeslagen, bijvoorbeeld verkeerde naamgeving voor gegevens, lokaal opgeslagen (in variabelen). Gegevens worden dubbel opgeslagen. Nummers zijn als string opgeslagen. Meerdere waarden die betrekking hebben op dezelfde informatie-eenheid (bijv. stationsnamen) worden niet gegroepeerd opgeslagen in een list, tuple of dictionary.
veelvoorkomende datatypes (number, string, dictionary, list, tuple) op correcte wijze kunnen gebruiken en daar bewerkingen op kunnen toepassen met behulp van operators.	10	Er wordt gebruik gemaakt van list-in-list of dict-in-dict waar dat kan. Statische informatie wordt opgeslagen als tuple.	Wanneer er 'lookups' gedaan moeten worden, wordt er gebruik gemaakt van een dictionary.	Waarden die betrekking hebben op dezelfde informatie-eenheid (bijv. meerdere stationsnamen) worden gegroepeerd opgeslagen in een list, tuple of dictionary	
op een correcte manier gebruik maken van de scope (globaal of lokaal) van variabelen en functies (namespaces). Hierbij is een basale kennis van objecten en klassen benodigd.	5	Functies zijn onderscheiden in objecten en klassen (encapsulation). Er is correct gebruik gemaakt van namespaces, en functies en variabelen worden hergebruikt waar dat kan. Er is sprake van information hiding: informatie wordt niet gedeeld tenzij dat nodig is.	Er is correct gebruik gemaakt van namespaces, en functies en variabelen worden hergebruikt waar dat kan. Er is sprake van information hiding: informatie wordt niet gedeeld tenzij dat nodig is.	Er wordt correct gebruik gemaakt van namespaces. Variabelen worden hergebruikt waar dat kan.	Variabelen worden niet hergebruikt waar dat kan.
de flow van een programma controleren door expressies correct te gebruiken in beslispunten (selection) of loops (iteration).	15	Er is recursie gebruikt als dat kan. Er is correct gebruik gemaakt van condities, waarbij alle mogelijke situaties zijn afgedekt. Er is qua iteraties gebruik gemaakt van while statements als dat efficiëntere code oplevert. Er is gebruik gemaakt van loop-optimalisatie, zie <a href="https://www.python.org/doc/essays/list2str/">https://www.python.org/doc/essays/list2str/</a>	Er is correct gebruik gemaakt van condities, waarbij alle mogelijke situaties zijn afgedekt. Er is qua iteraties gebruik gemaakt van while statements als dat efficiëntere code oplevert. Er is gebruik gemaakt van loop-optimalisatie, zie <a href="https://www.python.org/doc/essays/list2str/">https://www.python.org/doc/essays/list2str/</a>	Er is correct gebruik gemaakt van condities, waarbij alle mogelijke situaties zijn afgedekt. Er is qua iteraties gebruik gemaakt van while statements als dat efficiëntere code oplevert.	Iteraties worden niet of verkeerd gebruikt. Een commando wordt bijvoorbeeld vijf keer herhaald, terwijl dit in een for-loop kan. Conditioes worden verkeerd gebruikt, stringrepresentaties van een nummer worden bijvoorbeeld vergeleken met een int/float. Complexe condities geven een verkeerd resultaat, omdat de haken verkeerd staan (order of precedence).
weten wat source control is, en dat er verschillende tools zijn waarmee je dit kunt bewerkstelligen	5	Er is gewerkt met Git. Merges zijn aangebracht.	Er is gewerkt met Git.	Er is versiebeheer toegepast in de vorm van lokale opgeslagen versies van bestanden. Er is bijvoorbeeld gewerkt met Dropbox of Google Drive.	Er is geen versiebeheer toegepast.
een programma testen op correcte en logische werking (bijv. doctest) en kan omgaan met onverwacht gedrag in de code (debugging).	5	De code draait foutloos, waarmee we ervan uitgaan dat de code een debugging-slag heeft ondergaan. Bij doctesting zijn alle situaties afgedekt, bij nummer input bijvoorbeeld -groot getal, - klein getal, 0, + klein getal, en + groot getal, string input.	De code draait foutloos, waarmee we ervan uitgaan dat de code een debugging-slag heeft ondergaan. Bij doctesting zijn veel, zoniet alle situaties afgedekt.	De code draait foutloos, waarmee we ervan uitgaan dat de code een debugging-slag heeft ondergaan. Doctesting is aanwezig, maar niet alle situaties zijn afgedekt.	Er is geen of slechte doctesting toegepast. De code draait niet foutloos.
constructief feedback geven én ontvangen ten behoeve van een beter eindresultaat van het team (bijv. code review).	10	De sterktes en zwaktes zijn voor het hele team en per teamlid beschreven (feed back). Er worden mogelijke oplossingen beschreven (feed forward). Er heeft minstens één keer onderlinge code review plaats gevonden.	De sterktes en zwaktes zijn voor het hele team en per teamlid beschreven (feed back). De beschrijvingen zijn helder, en voorzien van duidelijke voorbeelden. Er zijn docstrings gebruikt om het programma en haar functies toe te lichten. De werking en output van een functie is correct beschreven.	De sterktes en zwaktes voor het team en de teamleden zijn beschreven, maar dit is erg summier beschreven. Er zijn geen of inadequate oplossingen aangedragen.	Delen van de constructieve feedback op het team of teamleden ontbreken, en er worden inadequate oplossingen aangedragen.
de werking van een programma toelichten door het gebruik van duidelijke en taalkundig juiste Python docstrings.	5	De Python docstring convention is aangehouden, zie <a href="https://www.python.org/dev/peps/pep-0257/">https://www.python.org/dev/peps/pep-0257/</a>		Er zijn docstrings gebruikt, maar deze zijn niet duidelijk, of bevatten te weinig informatie.	Het programma is niet of slecht gedocumenteerd.

zijn eigen Python ontwikkelomgeving inrichten waarbij gebruik wordt gemaakt van een Integrated Development Environment (IDE)	0				
bij het werken in een team een duidelijke planning en taakverdeling opstellen.	5	Er is een duidelijke rol- of taakverdeling aanwezig in het verslag. Iedereen heeft zijn/haar verantwoordelijkheid voor een bepaald onderdeel van de applicatie. Er mogen verschillen zijn in de hoeveelheid programmacode die per teamlid oplevert, maar dit wordt opgevangen door in meer of mindere mate betrokken zijn bij het proces van analyse van het probleem, modelleren (bijv. PSD's) en de documentatie. Als de planning niet klopt, dan is deze waarneembaar bijgesteld en onderbouwd (waarom is de planning bijgesteld?)	Er is een duidelijke rol- of taakverdeling aanwezig in het verslag. Iedereen heeft zijn/haar verantwoordelijkheid voor een bepaald onderdeel van de applicatie. Er mogen verschillen zijn in de hoeveelheid programmacode die per teamlid oplevert, maar dit wordt opgevangen door in meer of mindere mate betrokken zijn bij het proces van analyse van het probleem, modelleren (bijv. PSD's) en de documentatie.	Er is een rol- of taakverdeling aanwezig in het verslag, maar deze is te summier om te bepalen wie welk onderdeel van het project voor zijn/haar rekening heeft genomen. Als de planning niet klopt, dan is deze waarneembaar bijgesteld.	Er is geen duidelijke rol- of taakverdeling aanwezig in het verslag. De planning is niet gehaald, of de planning is niet waarneembaar bijgesteld.
aangeven welke stappen zij ondernomen heeft om een bepaald programmeerconcept onder de knie te krijgen, en welke stappen hierin succesvol waren en welke minder goed werkten (zelfreflectie).	10	Er is een beknopte, maar goede zelfreflectie geschreven volgens de STARR-methode. Verder bevat het verslag een goede beschrijving van de stappen die ondernomen zijn. Zo kloppen de PSD's en komen ze overeen met de geprogrammeerde code.	Er is een reflectieverslag geschreven volgens de STARR-methode en dit bevat een goede beschrijving van de stappen die ondernomen zijn. Zo kloppen de PSD's en komen ze overeen met de geprogrammeerde code.	Er is een reflectieverslag geschreven, maar niet volgens de STARR-methode (of iets vergelijkbaars) en/of dit bevat een slechte beschrijving van de stappen die ondernomen zijn. Zo kunnen PSD's (of vergelijkbare diagrammen) ontbreken, kloppen de PSD's niet, of komen ze niet overeen met de geprogrammeerde code.	Er is geen reflectieverslag geschreven, of meerdere delen van het verslag ontbreken of zijn niet voldoende.
	100				