



Co-funded by the  
Erasmus+ Programme  
of the European Union



2020-1-UK01-KA227-YOU-094543

## IO1- A3: Digitaal ontwerpcompendium Module: Computer Programming

KA2 - Cooperation for innovation and the exchange of good practices  
Partnerships for creativity



The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



## REVISION HISTORY

Version	Date	Author	Description	Action	Pages
1.0	15/10/2021	CIVIC	Creation	C	12

(\* ) Action: C = Creation, I = Insert, U = Update, R = Replace, D = Delete

## REFERENCED DOCUMENTS

ID	Reference	Title
1	2020-1-UK01-KA227-YOU-094543	HerTour4Youth Proposal
2		

ID	Reference	Title
1	Deliverable IO1.A2	Elaboration of Training Methodology
2		

## APPLICABLE DOCUMENTS





## Contents

1. Introduction .....	4
1.1 Leerresultaten .....	4
1.2 Sleutelwoorden: .....	4
1.3 Geschatte zittijd .....	4
1.4 Verklarende woordenlijst .....	4
2. Computerprogrammering:.....	6
2.1.1 Programmeertalen .....	10
2.1.2 Programmeerproces .....	12
2.1.3 Programmeringsprincipes .....	16
2.1.4 Hoe kun je computerprogrammeur worden? .....	17
3. Beoordeling .....	21
3.1 kennisbeoordeling.....	21
3.2 Vaardigheidsbeoordeling .....	23
4. Referenties .....	25



# 1. Introduction

Als je je afvraagt wat computerprogrammering is, stel je dan een heel eenvoudige computermachine voor in je huis (het kan een thermostaat zijn, je verbindingsrouter enz.). Weet nu dat die computer niet alleen werkt, maar door iemand is geprogrammeerd. Die iemand is een computerprogrammeur. Elke computer heeft een reeks instructies nodig om soepel te kunnen functioneren. Dat is waar computerprogrammering over gaat: het kan bestaan uit een eenvoudige set instructies om specifieke acties te vergemakkelijken tot een complexe set instructies met betrekking tot het lezen en sorteren van gegevens.

## 1.1 Leerresultaten

Na het voltooien van deze module ben je in staat om:

- Een basiskennis te hebben van programmeerprincipes
- Een diepere kennis te hebben van hoe ICT-tools werken
- Programmeerprincipes kunnen toe te passen

## 1.2 Sleutelwoorden:

- ICT
- Computerprogrammeur
- Programmeertalen

## 1.3 Geschatte zittijd

1 uur.

## 1.4 Verklarende woordenlijst

**ICT:** Staat voor "Informatie- en Communicatietechnologieën". ICT verwijst naar technologieën die toegang bieden tot informatie via telecommunicatie. (TechTerms, 2021)

**Computerprogrammeur:** een computerprogrammeur is de persoon die verantwoordelijk is voor het maken van instructies die een computer moet uitvoeren



door code te schrijven en te testen waarmee applicaties en softwareprogramma's succesvol kunnen werken. (Techopedia, 2014)

**Programmeertalen:** computerprogrammeertalen zijn alle verschillende talen voor het uitdrukken van een reeks gedetailleerde instructies voor een digitale computer. (D. Hemmendinger, 2021)

**Analoog:** bijvoeglijk naamwoord dat verwijst naar een mechanisme of apparaat waarin informatie wordt weergegeven door continu variabele fysieke hoeveelheden. Het is in tegenstelling tot digitaal. (Merriam Webster, 2021)

**Kunstmatige intelligentie:** Kunstmatige intelligentie maakt gebruik van computers en machines om de probleemoplossende en besluitvormingscapaciteiten van de menselijke geest na te bootsen, zonder zich te beperken tot methoden die biologisch waarneembaar zijn." (IBM, 2020)

**Machine learning:** machine learning is een tak van kunstmatige intelligentie (AI) en informatica die zich richt op het gebruik van gegevens en algoritmen om de manier waarop mensen leren te imiteren, waarbij de nauwkeurigheid geleidelijk wordt verbeterd. (IBM, 2020)

**Binair:** Binair is een getsysteem met grondtal-2, uitgevonden door Gottfried Leibniz, dat uit slechts twee cijfers of cijfers bestaat: 0 (nul) en 1 (één). Dit nummeringssysteem is de basis voor alle binaire code, die wordt gebruikt om digitale gegevens te schrijven, zoals de computerprocessorinstructies die elke dag worden gebruikt. (ComputerHoop, 2021)



## 2. Computerprogramming:

### 2.1 Inleiding

In onze computergestuurde wereld weten we dat computers veel opmerkelijke dingen kunnen doen.

De snelle technologische vooruitgang die we de afgelopen decennia hebben gezien, heeft een verrijkend effect op hoe alles werkt. Je moet onder de oppervlakte kijken om te zien waar de belangrijkste veranderingen hebben plaatsgevonden. Computers vervingen miljoenen werkuren en magazijnen van analoge machines door snellere, veiligere en betrouwbaardere systemen.

Omdat computers op code draaien, is het duidelijk waarom je het overal kunt vinden. Computers zullen verouderde technologie blijven vervangen in alles, van magnetrons tot energiecentrales. En de aanwezigheid van code in ons dagelijks leven zal toenemen.

Ook heeft bijna iedereen tegenwoordig een laptop die kan worden gebruikt voor eenvoudige computerprocessen, zoals het maken van een spreadsheet of het maken van eenvoudige documenten. Er zijn echter ook mensen die hun laptop op een complexere en professionelere manier gebruiken. Dat kan nodig zijn om miljoenen financiële transacties per dag af te ronden en om de infrastructuur te controleren die het moderne leven mogelijk maakt.

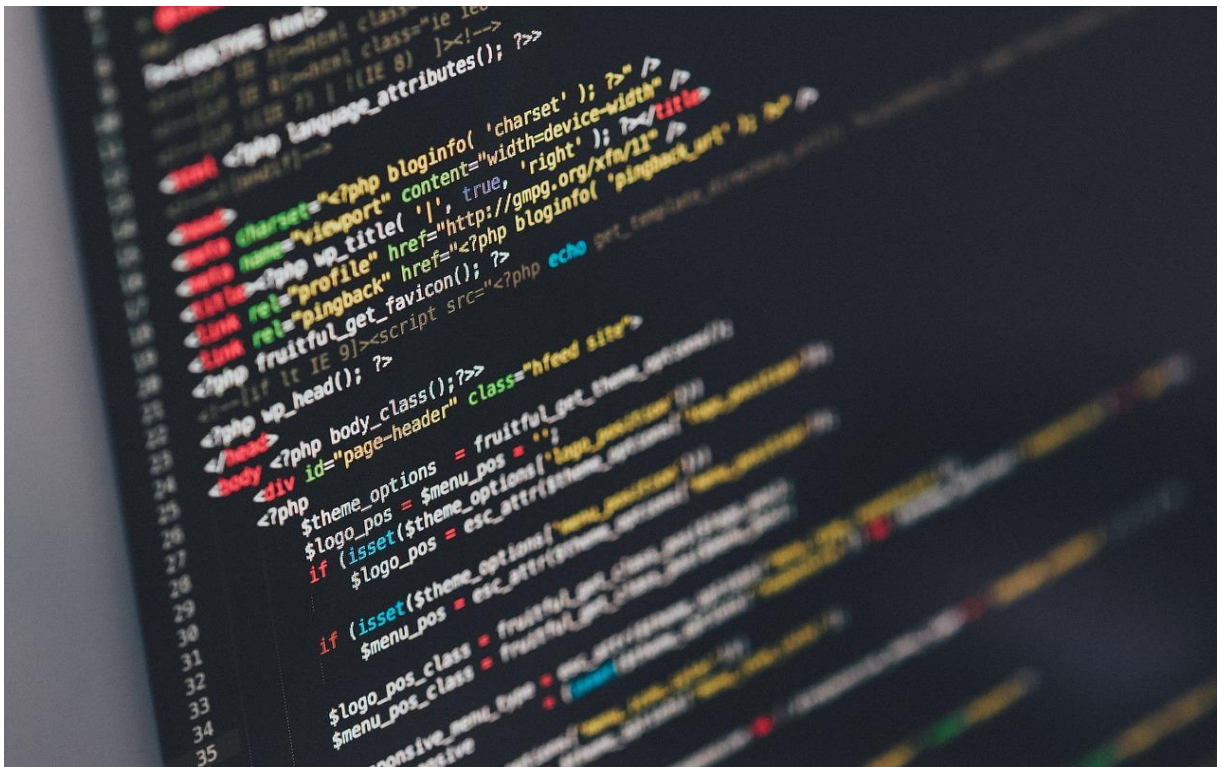
Dat gezegd hebbende, is het belangrijk om te onthouden dat geen enkele computer iets kan doen als het niet door een mens is geprogrammeerd. In dit geval hebben we het over de figuur van de computerprogrammeur.



De computerprogrammeur is de professionele figuur die verantwoordelijk is voor het schrijven van codes, of codering, die instrueren hoe een computer, applicatie of softwareprogramma zou moeten presteren.

Zoals eerder vermeld, kan deze reeks instructies vrij eenvoudig zijn en kan het hele coderingsproces slechts twee cijfers vereisen, maar het kan ook ingewikkelder zijn, zoals het lezen van gegevens van temperatuursensoren om een thermostaat aan te passen, het sorteren van gegevens om ingewikkelde planning te voltooien of kritische rapporten of spelers door gelaagde werelden en uitdagingen in games leiden.

Terwijl informatica zich bezighoudt met theoretische ideeën op hoog niveau, is bijna elk aspect van het moderne leven afhankelijk van codering. Elke applicatie op onze telefoon, tablet of computer gebruikt computertalen om te draaien.



Bron: <https://pixabay.com/photos/code-coding-computer-data-1839406/>

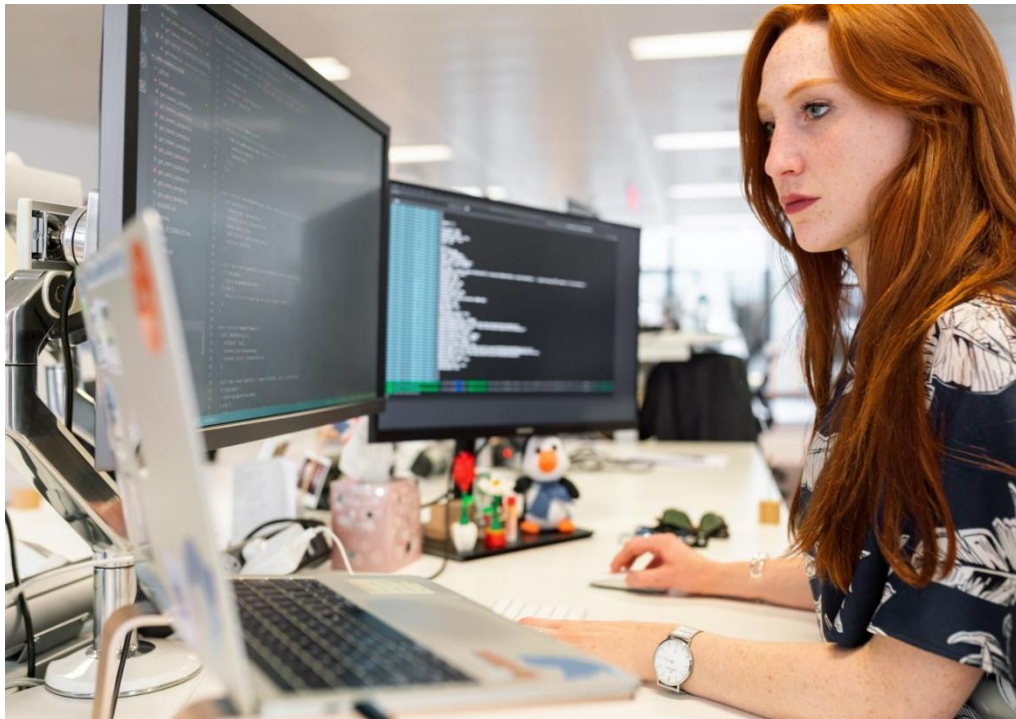
Andere digitale systemen zoals smart-tv's en rekenmachines gebruiken het ook.



Vrijwel elke nieuwe auto gebruikt het om alles te regelen, van airconditioningsystemen tot brandstofinjectoren. Steden gebruiken computers om verkeerslichten te bedienen.

Systemen die vroeger analoog waren, worden nu gestroomlijnd met behulp van geautomatiseerde systemen. Hierdoor kunnen ingenieurs een efficiënter en goedkoper systeem, structuur en machine bouwen. Verder gebruiken enkele van de meest geavanceerde technische gebieden (zoals kunstmatige intelligentie en machine learning) codering..<sup>1</sup>

## 2.2 Computerprogrammeur



Bron: <https://unsplash.com/photos/64YrPKiguAE>

---

<sup>1</sup> Weinstein, J. (2021, January 24). What Is Coding? Coding Definition and Uses. Career Karma. <https://careerkarma.com/blog/what-is-coding-used-for/>



Zoals eerder vermeld, is een computerprogrammeur de persoon die verantwoordelijk is voor het maken van instructies voor een computer om uit te voeren door code te schrijven en te testen waarmee applicaties en softwareprogramma's succesvol kunnen werken.

Een computerprogrammeur kan ook worden aangeduid als een programmeur, codeur, ontwikkelaar of software-ingenieur. De term wordt ook vaak gebruikt om te verwijzen naar een stand-alone softwareontwikkelaar, ontwikkelaar van mobiele applicaties, webontwikkelaar, softwareanalist, embedded firmware-ontwikkelaar, enzovoort.

Een computerprogrammeur is een ervaren professional die de uitgebreide instructies codeert, test, debugt en onderhoudt die bekend staan als computerprogramma's die computers moeten volgen om hun specifieke functies uit te voeren<sup>2</sup>

Some of the common tasks a computer programmer has to master comprise:

- Testing software performance.
- Resolving computer software problems.
- Modifying software programs to improve performance.
- Writing computer programming code.
- Collaborating with others to resolve information technology issues.

Het is gebruikelijk om computerprogrammeurs in twee typen in te delen: systeemp programmeurs en applicatieprogrammeurs.

---

<sup>2</sup> Techopedia. (2014, July 30). Computer Programmer. Techopedia.Com.  
<https://www.techopedia.com/definition/6589/computer-programmer>

Beide rollen coderen. Het verschil tussen de twee is dat applicatieprogrammeurs coderen om een bepaalde taak te beheren, zoals het coderen van een programma om de voorraad binnen een bedrijf te bewaken.

Aan de andere kant coderen systeemprogrammeurs programma's om systeemsoftware te onderhouden en te controleren, inclusief databasebeheersystemen en besturingssystemen (OS's).

### 2.1.1 Programmeertalen

Net als op mensen gebaseerde talen, hebben computers ook hun eigen talen. Er bestaan inderdaad meerdere computerprogrammeertalen die programmeurs kunnen gebruiken om met een computer te communiceren om een reeks specifieke taken uit te voeren.

Het gedeelte van de taal dat een computer kan begrijpen, wordt een 'binair' genoemd. Het vertalen van programmeertaal naar binair staat bekend als "compileren". Elke programmeertaal heeft zijn eigen kenmerken. Soms zijn er echter overeenkomsten tussen hen<sup>3</sup>.

Enkele van de meest voorkomende computerprogrammeertalen zijn:

- C
- SQL
- PHYTON
- JAVA
- R

---

<sup>3</sup> Writers, S. (2021, September 27). Guide to Programming Languages | ComputerScience.org. Get an Education the World Needs | ComputerScience.Org. <https://www.computerscience.org/resources/computer-programming-languages/>

- HTML<sup>4</sup>



Bron: <https://pixabay.com/illustrations/programming-languages-icon-898961/>

<sup>4</sup> Gallagher, J. (2021, May 4). The Most Popular Programming Languages. Career Karma. <https://careerkarma.com/blog/top-programming-languages-2021/>



Veel programmeertalen zijn relatief eenvoudig, maar ze doen verschillende dingen. Een van de meest populaire talen, JavaScript, wordt bijvoorbeeld voornamelijk gebruikt voor webpagina's en front-end ontwikkeling. Aan de andere kant wordt Python gebruikt voor zowel complete softwareprogramma's als websites.

Wat een softwareontwikkelaar ook kiest, is meestal aan hun discretie, omdat meer dan één programmeertaal vaak voor hetzelfde doel kan werken.

Het kennen van een of meer van deze programmeertalen is essentieel als we een carrière willen opbouwen, niet alleen als computerprogrammeur, maar ook als software-ingenieur of elke baan op het gebied van datawetenschap waarvoor hightechkennis vereist is.

### 2.1.2 Programmeerproces

Het hoofddoel van programmeren is over het algemeen om een oplossing te bedenken om een probleem op te lossen. Het probleem kan iets groots zijn op internationale schaal of gewoon iets kleins om ons van verveling te verlossen.





Bron: <https://pixabay.com/illustrations/programming-languages-icon-898961/>

Het ontwikkelen van een programma omvat stappen die vergelijkbaar zijn met elke probleemoplossende taak. Er zijn vier hoofdstappen in het programmeerproces:

- Het **probleem definiëren**: in dit stadium probeert u het probleem te identificeren. Het zou echter verstandig zijn om te zeggen dat wat u echt probeert te identificeren de oplossing van het probleem is, aangezien u probeert te definiëren wat u wilt bereiken.

Het eerste dat u moet doen, is het analyseren van de vereisten. U moet inderdaad weten wat uw programma moet doen.

Het tweede wat u dan moet doen, is de lijst met vereisten in acht nemen en beslissen wat uw oplossing precies moet doen om hieraan te voldoen. Soms vereist een enkel probleem meerdere oplossingen, dus het is belangrijk om erachter te komen wat de beste oplossing is voor elk probleem.

U probeert dus nogmaals te specificeren welke oplossing u voor uw probleem moet kiezen.

- De **oplossing plannen** en ontwerpen: dit is over het algemeen de moeilijkste taak van allemaal, omdat je moet bedenken hoe je die specificaties en vereisten omzet in een werkend en functioneel programma.

Het ontwerpen van een oplossing betekent op een hoger niveau alle stappen beschrijven die de computer moet volgen om goed te kunnen functioneren of een specifieke taak uit te voeren.

Deze taak is om verschillende redenen het moeilijkst: u moet bijvoorbeeld wat meer leren over de functies van uw computer en de door u gekozen programmeertalen om te zien welke dingen gemakkelijk of moeilijk kunnen zijn; ook zoals eerder vermeld, kan een probleem verschillende oplossingen vereisen en zodra een oplossing is geïdentificeerd, is het belangrijk om al zijn sterke en zwakke punten te analyseren en de juiste keuze te maken.

Zodra het moeilijke deel is voltooid, moet uw ontwerpgedeelte echter worden voltooid. Dat betekent dat u een heel duidelijk idee heeft van hoe de computer aan uw specificatie zal voldoen, aan uw eisen zal voldoen en uiteindelijk uw oorspronkelijke probleem zal oplossen.

- Het **programma coderen**: In deze taak moet u uw computer instrueren hoe u uw ontwerp moet uitvoeren. Deze fase omvat 3 fasen:
  - **Coderen**: Coderen betekent het vertalen van uw ontwerp naar een functioneel programma, met behulp van een programmeertaal. In deze fase gaat u praktisch



zitten en begint u codes op uw computer te typen. Net als een essay, vereist het coderen van een programma dat je dingen zoals titels, inhoudspagina, inleiding en referenties opneemt. Als u klaar bent met uw codering, moet u deze indienen bij de computer om te zien hoe de computer reageert en wat ervan is.

- **Compileren:** in de compileerfase moet u het geschreven programma omzetten in een taal die de computer begrijpt, vandaar de binaire machinecode. Er zijn een aantal programma's genaamd compilers die dit voor u kunnen doen. Je moet echter heel precies zijn bij het schrijven van je programma in een compiler, omdat ze elke minimale fout zullen detecteren.
- **Debugging:** zoals eerder vermeld, zal de compiler elke fout in het programma detecteren. Maak je geen zorgen, dat is een veel voorkomend iets. Daarom is deze fase er. Als je eenmaal alle fouten in je programma hebt geïdentificeerd, kijk dan, dankzij de hulp van de compilers, nog eens naar het originele programma, identificeer de fouten, corrigeer de code en compileer deze opnieuw. Deze cyclus van code -> compileren -> debuggen zal vaak vele malen worden herhaald voordat de compiler er tevreden mee is en je zult eindelijk een programma hebben dat werkt. Dit kan even duren.
- Je kunt er ook voor kiezen om kleine delen van je programma te schrijven, ze te compileren en te debuggen, in plaats van je in één keer op het hele programma te concentreren.
- **Het programma testen:** Het laatste deel van het programmeerproces omvat het testen van uw creatie en het controleren of het precies werkt zoals u het wilde. Ik weet dat je misschien denkt dat de compiler je al het juiste programma heeft gegeven, maar dat betekent niet dat de juiste code kan oplossen wat je oorspronkelijke probleem was. Dat is aan u om te controleren.



Als er eventuele fouten worden gevonden, is de enige optie om het programma opnieuw te bezoeken, de fout te herstellen door de code te wijzigen en deze opnieuw te compileren. Denk eraan om voorzichtig te zijn bij het wijzigen van iets in de code, omdat het veranderen van een klein ding andere dingen in het hele programma kan beïnvloeden. Deze fase kan worden vergeleken met een andere debugging-fase. Eindelijk, na opnieuw compileren, fouten herstellen, debuggen en testen, en nadat je ervoor hebt gezorgd dat je programma werkt volgens je eisen en specificaties, heb je een oplossing voor je probleem!<sup>5</sup>

### 2.1.3 Programmeringsprincipes

Op dit moment weten we dat een van de belangrijkste kenmerken van computerprogrammering het schrijven van codes is. Of u nu over het algemeen alleen of in een team werkt, onthoud dat uw codes gemakkelijk te lezen en te onderhouden moeten zijn voor andere mensen.

Dat is de reden waarom enkele basisprogrammeerprincipes die elke programmeur zou moeten kennen om eenvoudige codes te schrijven, met schone variabelen en in staat om stand te houden, zelfs na testen en eventuele wijzigingen.

Enkele van deze principes zijn:

- **KISS: Keep It Simple, Stupid:** Het KISS-principe is een van de belangrijkste principes om naar te leven in de programmeerwereld. Het betekent dat je code zo eenvoudig mogelijk moet schrijven. U hoeft niet te proberen te pronken met geavanceerde code. Als je het in één regel kunt schrijven, gebruik dan één regel. En onthoud dat sommige codes na maanden opnieuw moeten worden bekeken, dus houd daar rekening mee en maak het eenvoudig.

---

<sup>5</sup> The Programming Process. (2020). Cs.Bham.Ac.Uk.  
<https://www.cs.bham.ac.uk/%7Erxb/java/intro/2programming.html>





- **DRY: Don't Repeat Yourself:** dit principe helpt je een veel voorkomende fout bij het coderen te onthouden, vandaar herhalingen. Vermijd het dupliceren van gegevens en logica.
- **Open/Gesloten:** dit principe zorgt ervoor dat u eraan denkt uw codes open te houden voor extensies, maar gesloten voor wijzigingen. Deze regel is handig wanneer u een bibliotheek met services aanbiedt die door anderen kunnen worden gebruikt. In zeer eenvoudige woorden, als iemand anders uw code wijzigt, zal de code breken of iets anders in de code wordt beïnvloed en zal niet werken zoals het normaal zou moeten. Daarom moet u alleen codes vrijgeven die directe wijziging voorkomen en uitbreiding aanmoedigen.
- **Enkele verantwoordelijkheid:** dit principe stelt dat elke klasse of module in een programma slechts één specifieke functionaliteit mag bieden. Daarom wordt de code op deze manier eenvoudig gehouden en wordt een ingewikkelder debugging-proces vermeden; Ten tweede wordt het moeilijker om extra functionaliteit te creëren voor een specifieke module.
- **YAGNI:** Je zult het niet nodig hebben: dit principe stelt dat je nooit moet proberen codes te schrijven voor een functionaliteit die je in de toekomst misschien nodig hebt. Dat zou betekenen dat je een probleem probeert op te lossen dat niet bestaat.

## 2.1.4 Hoe kun je computerprogrammeur worden?



Als computerprogrammeur kun je een baan vinden in de industrie voor het ontwerpen van computersystemen en aanverwante diensten. Veel informaticagebieden vereisen echter de figuur van een computerprogrammeur.

Deze omvatten bijvoorbeeld:

- Software-engineering
- Softwareontwikkeling
- Computertechniek
- Computergraphics
- Kunstmatige intelligentie

Over het algemeen moeten computerprogrammeurs een universitair diploma in data- en informatica hebben, wat je de basis kan leggen voor het begin van je carrière. Veel programmeurs kunnen echter ook autodidactische enthousiastelingen zijn met een aanhoudende interesse in programmeren.

Of je nu besluit om je carrière te beginnen met een universitaire opleiding of als zelfliefhebber, het belangrijkste is je kennis van de programmeertalen te verbeteren, geïnteresseerd te zijn in computerprogrammering en ook op de hoogte te blijven en nooit te stoppen met leren, want computerprogrammeren is een constante up-to-date discipline.





Bron: <https://pixabay.com/photos/scrabble-board-game-game-4370255/>

Een ander belangrijk ding dat veel banen voor computerprogramming vereisen, zijn professionele en non-profit professionele certificeringen die beschikbaar zijn.

Sommigen van hen omvatten:

- CISCO – Certified Network Associate, Certified Network Profession Routing and Switching, Certified Network Associate Security Credential
- Microsoft – Certified Solution Developer voor webapplicaties, Certified Solutions Associate Windows Server
- Beroepsverenigingen – Certificering van Software Development Associate, Comptia's Security+, Comptia's A+ Certificering, Comptia's Linux+



- Non-profit – Certified Information Systems Security Professional, Certified Information Security Manager, Certified Secure Software Lifecycle Professional Credential<sup>6</sup>

---

<sup>6</sup> Cote, J. (2021, August 19). What is Computer Programming? How to Become a Computer Programmer. SNHU.Edu. <https://www.snhu.edu/about-us/newsroom/stem/what-is-computer-programming>



## 3. Beoordeling

### 3.1 kennisbeoordeling

Quiz-achtige beoordeling op basis van de hoofdinhoud. Markeer het juiste antwoord indien nodig vet. Voeg 10 vragen toe voor uw module. Verhoog geleidelijk de moeilijkheidsgraad.

Vraag 1: De computerprogrammeur is de professionele figuur die verantwoordelijk is voor het schrijven van codes, of codering, die instrueren hoe een computer, applicatie of softwareprogramma zou moeten presteren.

[**Waar** onwaar]

Vraag 2: Het coderingsproces kan alleen een computer laten werken.

[Waar **onwaar**]

Vraag 3: Welke taken moet een computerprogrammeur beheersen?

[Problemen met computersoftware oplossen] [Softwareprogramma's aanpassen om de prestaties te verbeteren] [Computerprogrammeercode schrijven] [Softwareprestaties testen] [**Al het bovenstaande**]

Vraag 4: Het deel van de taal dat een computer kan begrijpen, wordt een "binair" genoemd. Hoe wordt het proces van het vertalen van programmeertaal naar binair genoemd?

[Coderen] [**Compileren**] [Debuggen]

Vraag 5: Wat is de juiste volgorde van de programmeerprocestaken?



[Het programma coderen, het programma testen, het probleem definiëren, de oplossing plannen en ontwerpen]

[De oplossing plannen en ontwerpen, het probleem definiëren, het programma coderen, het programma testen]

**[Het probleem definiëren, de oplossing plannen en ontwerpen, het programma coderen, het programma testen]**

Vraag 6: Wat zijn de drie fasen van de codeerfase binnen het programmeerproces?

[Testen] [**Coderen**] [Systematiseren] [**Compileren**] [**Debuggen**]

Vraag 7: Waaraan herinneren programmeerprincipes programmeurs om te doen bij het coderen? [om codes te gebruiken die andere programmeurs in de toekomst niet kunnen wijzigen] [om complexe codes te gebruiken die de functies verbeteren] [**om eenvoudige codes te schrijven**] [**om schone variabelen te gebruiken die sterk blijven na het testen**]

**[om eenvoudige codes te schrijven] [om schone variabelen te gebruiken die sterk blijven na het testen]**

Vraag 8 (matching): Verbind de termen met hun definities.

Binair: een door Gottfried Leibniz uitgevonden getalsysteem met grondtal-2 dat uit slechts twee cijfers of cijfers bestaat: 0 (nul) en 1 (één).

ICT: Dit woord verwijst naar technologieën die toegang bieden tot informatie via telecommunicatie.

Machine Learning: een tak van kunstmatige intelligentie (AI) en informatica die zich richt op het gebruik van gegevens en algoritmen om de manier waarop mensen leren te imiteren, waarbij de nauwkeurigheid geleidelijk wordt verbeterd

Analoog: bijvoeglijk naamwoord dat verwijst naar een mechanisme of apparaat waarin informatie wordt weergegeven door continu variabele fysieke hoeveelheden.

Het is in tegenstelling tot digitaal.

Vraag 9 (matching): Verbind de begrippen met hun uitleg.



Computerprogrammeur: persoon die verantwoordelijk is voor het maken van instructies voor een computer om uit te voeren door het schrijven en testen van code waarmee applicaties en softwareprogramma's succesvol kunnen werken.

Compileren: proces van het vertalen van programmeertaal naar binair.

Debuggen: proces waarbij alle bugs in het programma worden geïdentificeerd, dankzij de hulp van de compilers, het originele programma wordt beoordeeld, de fouten worden geïdentificeerd, de code wordt gecorrigeerd en opnieuw wordt gecompileerd.

Coderen: proces van het vertalen van het ontwerp naar een functioneel programma met behulp van een programmeertaal.

Testen: proces waarbij het programma wordt beoordeeld om te controleren of het precies werkt zoals u dat wilt.

### 3.2 Vaardigheidsbeoordeling

Analytisch denken is een belangrijke eigenschap van een programmeur. Het wordt ook wel analytisch redeneren, abstract denken of kritisch denken genoemd. Mensen die logisch kunnen denken, kunnen problemen analyseren en oplossingen bedenken. Dit is niet alleen waardevol bij het ontwikkelen van programma's, maar is van vitaal belang voor elke situatie die rationeel nadenken vereist. Mensen die logisch zijn:

- analyseren informatie of bronnen met betrekking tot een taak
- letten goed op wat er gebeurt
- bestuderen informatie objectief om te bepalen of het relevant of waar is
- focussen op feiten, niet op emoties
- ontwikkelen oplossingen voor problemen op basis van feiten
- schetsen ideeën duidelijk door ze in delen op te splitsen



- letten op details
- testen de effectiviteit van een oplossing en voeren herzieningen uit

Er zijn veel oefeningen en activiteiten om deze vaardigheid te oefenen. Sommige zijn heel eenvoudig, zoals boeken lezen, hersenspelletjes spelen of lid worden van een debatclub om je hersenen gestimuleerd te houden.

In het geval van een programmeur is een gemakkelijke manier om uw analytische denkvaardigheden te verbeteren, deze 5-stappengids te volgen en in gedachten te houden:

1. Analyseer het probleem: omvat het verzamelen van informatie, het onderzoeken van bronnen en het vaststellen van hiaten in vaardigheden of kennis;
2. Formuleer een plan: brainstorm over ideeën en organiseer je gedachten in een plan;
3. Ontwikkel code om het probleem op te lossen: begin met het schrijven van de code (kennis van computertalen is essentieel);
4. Evalueer de oplossing en herzie de code: bekijk de oplossing en identificeer indien nodig verbeterpunten;
5. Rechtvaardig beslissingen: presenteer bewijs dat verklaart waarom het programma een aanvaardbare oplossing is.

Meer informatie hier:

<https://www.technokids.com/blog/teaching-strategies/its-easy-to-improve-logical-thinking-with-programming/>





## 4. Referenties

Cote, J. (2021, August 19). What is Computer Programming? How to Become a Computer Programmer. SNHU.Edu. <https://www.snhu.edu/about-us/newsroom/stem/what-is-computer-programming>

Weinstein, J. (2021, January 24). What Is Coding? Coding Definition and Uses. Career Karma. <https://careerkarma.com/blog/what-is-coding-used-for/>

Writers, S. (2021, September 27). Guide to Programming Languages | ComputerScience.org. Get an Education the World Needs | ComputerScience.Org. <https://www.computerscience.org/resources/computer-programming-languages/>

Gallagher, J. (2021, May 4). The Most Popular Programming Languages. Career Karma. <https://careerkarma.com/blog/top-programming-languages-2021/>

Techopedia. (2014, July 30). Computer Programmer. Techopedia.Com. <https://www.techopedia.com/definition/6589/computer-programmer>  
The Programming Process. (2020). Cs.Bham.Ac.Uk.

<https://www.cs.bham.ac.uk/%7Erxb/java/intro/2programming.html>

