



**education**

---

Department:  
Education  
**REPUBLIC OF SOUTH AFRICA**

**NASIONALE KURRIKULUMVERKLARING  
(GRAAD 10-12)**

**PRAKTIESE ASSESSERINGSTAAK (PAT)**

**INLIGTINGSTEGNOLOGIE**

**2009**

Hierdie dokument bestaan uit 28 bladsye

# Inhoudsopgawe

Hierdie dokument bestaan uit drie afdelings:

<b>RIGLYNE VIR DIE ONDERWYSER .....</b>	<b>3</b>
<b>INSTRUKSIES VIR DIE LEERDER .....</b>	<b>1</b>
<b>ASSESSERINGSINSTRUMENTE .....</b>	<b>1</b>
FASE 1: KRITERIA EN PUNTETOEKENNING .....	2
FASE 2: KRITERIA EN PUNTETOEKENNING .....	3
FASE 3: KRITERIA EN PUNTETOEKENNING .....	5
FASE 4: KRITERIA EN PUNTETOEKENNING .....	9

# **Inligtingstechnologie**

## **Praktiese Assesseringstaak (PAT)**

### **2009**

## **Riglyne aan die onderwyser**

Hierdie afdeling bestaan uit 6 bladsye

## A Inleiding

Die doel van die PAT is om die onderwyser die geleentheid te gee om die toegepaste bevoegdheid van die leerder direk en sistematies waar te neem en te evalueer. Die PAT behels die toepassing van kennis, vaardighede en waardes in Inligtingstegnologie.

In Inligtingstegnologie tel die PAT 25% van die totale promosie-/sertifikaatpunt vir die vak (dit is 100 uit die 400 punte). Die word geimplimenter oor die eerste drie termyne van die skooljaar en moet as een uitgebreide taak onderneem word wat opgedeel word in verskillende fases of 'n reeks kleiner aktiwiteite.

Die IT PAT bestaan uit vier komponente:

- Navorsing en Analise
- Ontwerp
- Kodering en Implementering
- Dokumentasie en Algemene evaluering

## B Programmeringsprojek

### 1. Waaroor gaan die programmeringsprojek?

Die programmeringsprojek van die Praktiese Asseseringstaak in Inligtingstegnologie vereis dat die leerders 'n programmatuur-oplossing moet ontwikkel vir 'n spesifieke probleem in 'n gegewe scenario.

### 2. Beplanning van die programmingsprojek

#### 2.1 Beskrywing

##### Scenario vir die 2009 Graad 12 PAT:

Die skool het baie kleinsakeinitiatiewe waarvoor hulle graag programmatuur sal wil gebruik om dit te administreer. Byvoorbeeld

- 'n Snoepwinkel
- 'n Rekenaarherstelwinkel
- 'n Voorraadwinkel vir die skool (skooldrag, sportitems, ens.)
- 'n Boekwinkel of DVD-winkel, ens.

Sommige van jou mede-skoliere het moontlik ook programmatuur nodig vir kleinsakeondernemings wat skoolaktiwiteite te ondersteun. Byvoorbeeld

- 'n Foto-winkel wat fotos neem vir skoolaktiwiteite soos die Graad 12 afskeid
- 'n Rekenaarherstelwinkel wat die rekenaars van medeskoliere herstel
- 'n Klein drukkersbesigheid wat programme vir gebeurtenisse by die skool druk.

Identifiseer 'n toepassing wat benodig word in enige kleinsakeonderneming in die skoolomgewing waarvoor 'n program geskryf moet word.

Die program moet interaktief wees met 'n databasis en 'n grafiese koppelvlak (GUI) gebruik om met die gebruiker te kommunikeer. Java of Delphi moet gebruik word as

programmeringstaal. Die leerders moet saam met die program ook 'n tegniese handleiding en 'n gebruikershandleiding vir die program saamstel.

Die programmeringsprojek is oop. Dit beteken dat die leerder sy/haar eie onderwerp of fokus-area binne die konteks van die gegewe scenario mag kies.

Met die voltooiing van die projek moet die leerder die volgende vaardighede toepas:

- Navorsing (ondersoek)
- Analise
- Ontwerp
- Ontwikkeling van programmatuur
  - Programmeringsvaardighede deur die gebruik van die programmeringstaal wat bestudeer is
  - Ontwikkeling van 'n databasis
  - Ontwerp van 'n grafiese gebruikerskoppelvlak (GUI)

Gedeeltes van die evalueringinstrumente sal projekte penaliseer wat nie verwant is aan die gegewe scenario nie.

## 2.2 Ontwikkelingsfases van die projek

Die programmeringsprojek moet in 4 fases voltooi word soos aangedui in die volgende tabel:

Fases	Punte	%
Fase 1: Navorsing en Analise	30	12.0
Fase 2: Ontwerp	40	16.0
Fase 3: Kodering en Implementering	130	52.0
Fase 4: Dokumentering en Algemene evaluering	50	20.0
<b>Total</b>	<b>250</b>	<b>100</b>

**BELANGRIK:** Dokumentasie/bewys van wat die leerder gedurende elke fase gedoen het moet met gespesifiseerde intervalle ingehandig word. Dit is uiters belangrik dat bewyse voorsien word van die werk in al die fases. Die dokumente wat vereis word vir elke fase word verskaf in die leerderafdeling van hierdie dokument.

**Sperdatums** vir die inhandiging van die finale produk van elke fase sal deur die onderwyser vasgestel word. Die produk van elke fase sal ge-assesseer word en die punte sal aangeteken word.

(Sien ook Aanhangsel 1 in Leerprogramriglyne asook Aanhangsel 1 in die Vakassesseringsriglyne)

## 2.3 Vereistes van die projek

Die leerder moet aandag gee aan die volgende minimum kriteria:

### Navorsing en Analise

- 'n Beskrywing van die probleem in die leerder se eie woorde waarin die hoof aspekte uitgelig word.
- Navorsing van die onderwerp: Die navorsing sal bestaan uit inligting wat verkry is vanuit die werklike situasie waar die program gebruik gaan word
- Voorlopige ondersoek om die aard en die omvang van die probleem te identifiseer en om feite in te samel.
- Analisering van die probleem – Wat is die vereistes en wat moet die geprogrammeerde oplossing voorsien?

(Sien Leerderafdeling 2, Fase 1 sowel as die Asseseringsinstrument Fase 1)

### Ontwerp

- Ontwerp 'n oplossing – hoe sal die program/stelsel die verseistes aanspreek? Voorsien 'n goed beplande oplossing vir die probleem.

(Sien Leerderafdeling 2 Fase 2, Assesseringsinstrument Fase 2)

### **Kodering en Implementering**

- Die projek moet die belangrikste ontwikkelingsinstrumente naamlik databasisontwerp en programmering insluit op 'n geïntegreerde manier. (Ander toepassings kan saam met hierdie ontwikkelingsinstrumente geïntegreer word)
- Ander aspekte van die programmingsprojek wat ge-assesseer sal word:
  - Programmingstyl
  - Grafiese Gebruikerskoppelvlak (GUI)
  - Gebruik van Mens-Rekenaar Interaksie beginsels
  - Vlak van kundigheid wat vereis word en funksionaliteit van die program
  - Robuustheid van die program insluitend die gebruik van defensiewe programmingstegnieke
  - Of die projek ooreenstem met die oorspronklike mikpunte en doelwitte

(Sien Leerderafdeling 2 Fase 3, Assesseringsinstrument Fase 3)

### **Dokumentasie en algemene evaluering**

- Dokumenteer die oplossing, installasieprosedures asook apparatuur- en programmatuurvereistes – Tegnieke Handleiding
- Stel 'n gebruikershandleiding saam
- Evalueer die volgende:
  - Tydsbestuur van die leerder – Het hy/sy al die take betyds ingehandig?
  - Geskiktheid van die oplossing binne die konteks van die scenario.

(Sien Leerderafdeling 2 Fase 4, Assesseringsinstrument Fase 4)

## **3. Instruksies aan die leerders**

Verwys na Leerderafdeling van hierdie dokument.

## **4. Hulpbronne**

Die leerder sal die volgende hulpbronne nodig hê om die projek te voltooi:

- Toegang tot 'n rekenaar met die volgende programme:
  - Programmeringstaal: Java or Delphi
  - Woordverwerker soos MS Word
  - Databasisprogrammatuur soos MS Access
- IDE (vir Delphi is dit deel van die programmeringstaal maar vir Java word addisionele programmatuur benodig soos JBuilder/Turbo JBuilder/Netbeans/Eclipse/JCreator)

## **5. Assesering van die PAT**

Die projek moet onder gekontroleerde toestande in die klaskamer voltooi word en deur die onderwyser gefasiliteer en gemonitor word.

Sien Assesseringsinstrumentafdeling vir die assesseringsblaaie vir die verskillende fases.

Die onderwyser moet seker maak dat die leerders die volgende dokumente ontvang aan die begin van hulle Graad 12 jaar:

- Die “Instruksies aan die Leerders”-afdeling wat in hierdie dokument ingesluit is
- Die assesseringsinstrumente vir al die fases wat in hierdie dokument ingesluit is.

Leerders moet toegelaat word om nabetragting te hou oor die punte wat hulle gekry het en die foute wat hulle gemaak het aan te spreek voordat hulle die volgende fase voltooi, MAAR die punte wat toegeken is vir die aanvanklike evaluering sal NIE hersien word nie.

Daar sal van leerders verwag word om hulle stelsel te demonstreer vir 'n ondervragingsessie aan die einde van fase 3. Onderwysers moet die projekte evalueer volgens die assesseringsinstrument wat voorsien is vir fase 3.

Leerders sal **NIE** toegelaat word om die onderwerpe van hulle projekte te verander nadat fase 1 en 2 voltooi en ge-asseseeer is nie. Indien 'n leerder wel besluit om sy/haar onderwerp te verander nadat fase 1 of 2 reeds voltooi is, moet die leerder fase 1 en/of 2 weer oordoen vir die nuwe onderwerp. Die onderwyser sal die fases wat oorgedoen is NIE weer assesseer nie. Die punte vir die oorspronklike fases moet aangeteken word. Fase 3 sal nie ge-asseseeer word nie tensy fase 1 en 2 navorsing weerspieel wat vir die nuwe onderwerp gedoen is nie.

Die korrelasie tussen al die fases moet streng en voortdurend gekontroleer word tydens assessering sowel as tydens klustermoderering. Bewys van wat in die vorige fases gedoen is moet altyd beskikbaar wees gedurende assessering en moderering van 'n spesifieke fase van ontwikkeling.

#### **Riglyne vir die demonstrasie en interne evaluering:**

- Die onderwyser moet datums en tye skeduleer vir demonstrasies. Laat ongeveer 20 minute toe per projek asook 10 minute vir die opstelling en opsommende terugvoer van die onderwyser na die tyd.
- Die ontwikkeling van die projek is 'n aaneenlopende proses. Die onderwyser moet altyd kyk na die werk wat in vorige fases gedoen is wanneer 'n spesifieke fase van ontwikkeling ge-asseseeer word. Die onderwyser moet die vordering tydens die projek deeglik monitor in verband met die werk wat in vorige fases gedoen is.
  - Die vereistes wat geïdentifiseer is en uitgespel is in fase 1 moet gereflekteer word in fase 2 – die ontwerpfase.
  - Die werk wat gedurende die ontwerpfase – fase 2 – moet gereflekteer word in die kodering en implementering van die projek – fase 3.
  - Die dokumentasie wat in fase 4 gedoen is moet ooreenstem met die gekodeerde projek – fase 3.
- Die leerder moet al die vorige dokumentasie (fase 1 en fase 2) byderhand hê wanneer die demonstrasie vir fase 3 plaasvind.
- Die demonstrasies word elektronies op 'n rekenaar gedoen.
- Die leerder moet sy/haar program op die rekenaar uitvoer en al die funksies wat die program bied aan die onderwyser uitwys vir assessering.
- Die onderwyser moet die leerder vra om sekere van of al die toetsprosedures uit te voer om alle fasette van die program te toets
- Die onderwyser kan addisionele toetsprosedures laat uitvoer om seker te maak dat die hele program werk.

- Die onderwyser met die nasienblad vir Fase 3 gebruik om punte toe te ken tydens die demonstrasie.
- Die onderwyser moet **willekeurig 'n paar stukke programmeringskode (uitsluitend die 10% geleende kode) in die projek uitwys wat die leerder aan die onderwyser moet verduidelik.** Dit word gedoen om seker te maak dat alle kode deur die leerder self geskryf is. 'n Soortgelyke prosedure sal ook tydens eksterne moderering gevolg word. Indien die leerder nie op hoogte is van die kode wat in sy eie projek gebruik is nie, kan **geen punte vir die projek aan die leerder toegeken word nie.**
- Die leerder handig na afloop van die demonstrasie die elektroniese weergawe van die projek in sodat die onderwyser weer daarna kan kyk indien nodig voordat die finale punt toegeken word.

NOTA: Nadat die produk van 'n fase ingehandig is en ge-assesseer is, sal dit nie weer ge-assesseer word nie.

## 6. Rekordering en Rapportering

Vir elke fase sal die onderwyser die fase assesseer, die punt aanteken en terugvoer aan die leerder gee.

Die punte vir die verskillende fases sal bymekaar getel en verwerk word na 'n punt uit 100 wat die finale punt sal wees.

## 7. Riglyne vir die bestuur van die PAT

Daar is twee maniere waarop die programmingsprojek bestuur kan word:

### Opsie 1:

- Spandeer een of twee periodes per week aan die programmingsprojek terwyl daar skoolgehou word gedurende die res van die week se periodes om die Graad 12 kurrikulum af te handel. As hierdie opsie gevolg word, word daar ongeveer 14 - 16 weke benodig om die programmingsprojek te voltooi

### Opsie 2:

- Jy kan verkies om die Graad 12 kurrikulum grootliks af te handel en dan 'n aaneenlopende tydperk aan die programmeringsprojek te spandeer

Daar word aanbeveel dat die naam en die onderwerp van elke leerder aan die begin van die fase 1 aangeteken word om te voorkom dat "kits projekte" skielik opduik wat moontlik nie die leerder se eie werk kan wees nie. Onderwysers moet seker maak dat leerders projekte "registreer" wat hulle instaat sal wees om te voltooi om te voorkom dat onderwerpe verander word omdat die onderwerp te moeilik of te tydrowend is.

# **Inligtingstechnologie**

## **Praktiese Assesseringstaak (PAT)**

### **2009**

## **Instruksies aan die leerder**

Hierdie afdeling bestaan uit 11 bladsye

# Inligtingstegnologie – Praktiese Assesseringstaak (PAT)

Die IT PAT bestaan uit vier komponente:

- Navorsing en analise
- Ontwerp
- Implementering en Kodering
- Dokumentasie en algemene evaluering

## 1. Beplanning van die programmingsprojek

### 1.1 Beskrywing

#### Scenario vir die 2009 Graad 12 PAT:

Die skool het baie kleinsakeinitiatiewe waarvoor hulle graag programmatuur sal wil gebruik om dit te administreer. Byvoorbeeld

- 'n Snoepwinkel
- 'n Rekenaarherstelwinkel
- 'n Voorraadwinkel vir die skool (skooldrag, sportitems, ens.)
- 'n Boekwinkel of DVD-winkel, ens

Sommige van jou mede-skoliere het moontlik ook programmatuur nodig vir kleinsakeondernemings wat skoolaktiwiteite te ondersteun. Byvoorbeeld

- 'n Foto-winkel wat fotos neem vir gebeurtenisse soos die Graad 12 afskeid
- 'n Rekenaarherstelwinkel wat die rekenaars van medeskoliere herstel
- 'n Klein drukkersbesigheid wat programme vir gebeurtenisse by die skool druk.

Identifiseer 'n toepassing wat benodig word in enige kleinsakeonderneming in die skoolomgewing waarvoor 'n program geskryf moet word.

Die program moet kommunikeer met 'n databasis en met 'n grafiese koppelvlak (GUI) wat met die gebruiker kommunikeer. Java of Delphi moet gebruik word as programmeringstaal. Behalwe vir die program moet jy ook 'n tegniese handleiding en 'n gebruikershandleiding vir die program saamstel.

Met die voltooiing van die projek moet jy die volgende vaardighede toepas:

- Navorsing (Ondersoek)
- Analise
- Ontwerp
- Ontwikkeling van programmatuur
  - Programmeringsvaardighede deur die gebruik van die programmeringstaal wat bestudeer is
  - Ontwikkeling van 'n databasis
  - Ontwerp van 'n grafiese gebruikerskoppelvlak (GUI)

Gedeeltes van die evalueringsinstrumente sal projekte penaliseer wat nie verwant is aan die gegewe scenario nie.

Nota: Jou finale program moet een enkele logies-verwante program wees. Indien 'n projek uit twee of meer losstaande gedeeltes bestaan, sal slegs een van die losstaande gedeeltes as die projek beskou word en ge-assesser word.

## 1.2 Ontwikkelfases van die projek

Die programmingsprojek moet in 4 fases voltooi word soos aangedui in die volgende tabel:

Fases	Punte	%
Fase 1: Navorsing en Analise	30	12.0
Fase 2: Ontwerp	40	16.0
Fase 3: Kodering en Implementering	130	52.0
Fase 4: Dokumentering en algemene evaluering	50	20.0
<b>Total</b>	<b>250</b>	<b>100</b>

Dokumentasie/bewys van wat jy gedurende elke fase gedoen het moet met gespesifiseerde intervalle ingehandig word. Die bewys en afvoer van elke fase word hieronder bespreek. Punte kan slegs toegeken word as jy hierdie bewyse aan die onderwyser voorsien EN indien die werk wat ge-assesseer gaan word verband hou met wat in die vorige fases gedoen is.

**Sperdatums vir die inhandiging van die finale produk van elke fase sal deur die onderwyser vasgestel word.**

Bestudeer die assesseringsinstrument vooraf om seker te maak dat jy al die relevante vereistes aangespreek het volgens die assesseringsinstrumente.

Neem die terugvoer van die onderwyser op die assesseringsinstrument in ag en verbeter jou werk vir die volgende fase daarvolgens. Op 'n paar plekke word punte toegeken vir die verbetering van werk wat in vorige fases nie reg gedoen was nie.

Al die dokumentasie van vorige fases moet beskikbaar wees aan die onderwysers tydens elke assessering.

## 1.3 Hulpbronne wat benodig word vir die projek

Jy sal die volgende hulpbronne nodig hê om die projek te doen:

- Toegang tot 'n rekenaar met die volgende programme:
  - Programmeringstaal: Java or Delphi
  - Woordverwerker soos MS Word
  - Databasisprogrammatuur soos MS Access
- IDE (vir Delphi is dit deel van die programmeringstaal maar vir Java word addisionele programmatuur benodig soos JBuilder/Turbo JBuilder /Netbeans/Eclipse/JCreator)

Die projek moet onder **gekontroleerde toestande** in die klaskamer gedoen word en deur die onderwyser gefasiliteer en gemonitor word.

Jy moet aandag gee aan die volgende minimum kriteria:

### Navorsing en Analise

- 'n Beskrywing van die probleem in jou eie woorde waarin die hoof aspekte uitgelig word.
- Navorsing van die onderwerp: Die navorsing sal bestaan uit inligting wat verkry is van die werklike situasie waar die program gebruik gaan word

- Voorlopige ondersoek om die aard en die omvang van die probleem te identifiseer en om feite van potensiële gebruikers in te samel, insluitend hulle behoeftes en enige beperkings wat daar mag wees.
- Analisering van die probleem – Wat is die vereistes en wat moet die geprogrammeerde oplossing voorsien?

(Sien die volgende afdeling Fase 1 en die assesseringsinstrument van Fase 1)

### **Ontwerp**

- Ontwerp 'n oplossing – hoe sal die program/stelsel die vereistes aanspreek? Voorsien 'n oplossing wat goed beplan is en die vereistes aanspreek in terme van:
  - Toevoer, verwerking en afvoer
  - Die struktuur en inhoud van die databasis
  - GUI en die vloei van gebeure.

(Sien die volgende afdeling Fase 2 en die Assesseringsinstrument van Fase 2)

### **Kodering en Implementering**

- Die projek moet die belangrikste ontwikkelingsinstrumente naamlik databasisontwerp en programmering insluit op 'n geïntegreerde manier. (Ander toepassings kan met hierdie ontwikkelingsinstrumente geïntegreer)
- Ander aspekte van die programmingsprojek wat ge-assesseer sal word:
  - Programmingstyl
  - Grafiese Gebruikerskoppelvlak (GUI)
  - Gebruik van Mens-Rekenaar Interaksie beginsels
  - Vlak van kundige programmering
  - Funkisionaliteit van die program
  - Robuustheid van die program insluitend die gebruik van defensiewe programmeringstechnieke
  - Of die projek ooreenstem met die oorspronklike mikpunte en doelwitte

(Sien die volgende afdeling Fase 3 en die Assesseringsinstrument van Fase 3)

### **Dokumentasie en algemene evaluering**

- 'n Drukstuk van die bronkode, kommentaar ingesluit, installasieprosedures asook apparatuur- en programmatuurvereistes – Tegnieuse Handleiding
- Stel 'n gebruikershandleiding saam
- Demonstrasie van die finale produk en ondervraging deur die onderwyser.
- Die onderwyser sal die volgende evalueer
  - Tydsbestuur van die leerder – Het jy al die take betyds ingehandig?
  - Geskiktheid van die oplossing binne die konteks van die scenario.

(Sien die volgende afdeling Fase 4 en die Assesseringsinstrument van Fase 4)

## **2. Instruksies vir die fases van die programmingsprojek**

Die instruksies vir die verskillende fases is as volg:

**FASE 1:****Navorsing en Analise****Sperdatum:** \_\_\_\_\_

Met die voltooiing van hierdie fase moet jy agtergrondinligting opspoor omtrent jou onderwerp en bepaal *wat* die program/stelsel moet doen en voorsien:

## 1. Probleemstelling

- Beskryf/Verduidelik in jou eie woorde wat die probleem is wat opgelos moet word. Hierdie beskrywing moet nie 'n beskrywing van rekenaarkode wees of die oplossing nie. Al wat vereis word is 'n beskrywing van die probleem wat jy ondersoek in die werklike konteks of situasie soos dit geïdentifiseer is.

## 2. Besluit op 'n onderwerp vir jou program

- Doen navorsing oor die onderwerp/Ondersoek die onderwerp om feite in te samel en om die aard en die omvang van die probleem vas te stel
- Jy sal dalk die volgende wil uitvind:
  - Watter organisasies sal moontlik die programmatuur gebruik wat jy besig is om te ontwikkel
  - Watter stelsel of prosesse word gebruik in die afwesigheid van programmatuur (bv. papier-gebaseerde stelsel)
  - Inligting wat spesifiek verband hou met jou onderwerp (bv. as jy 'n program ontwikkel wat te doen het met boeke, dan moet jy inligting insamel oor boeke soos ISBN-nommers, uitgewers, ens.)
- Demografiese inligting oor die gebruikers van jou stelsel (ouderdom, geslag, gestremdhede, vlak van geletterdheid, ens)
- Jy moet 'n volledige lys van verwysings byhou van al die bronne (webblaaie, boeke) wat gebruik is.
- Bewys van navorsing: Goed-geformateerde en voorgestelde opsomming van al die bevindinge van die navorsing

## 3. Bepaal vereistes vir die program

- Hou besprekings met die eind-gebruikers, maak notas, versamel dokumente en maak opsommings en afleidings uit al hierdie inligting.
- Hou dop hoe die huidige stelsel werk. Indien moontlik, maak notas.
- Maak notas oor die toevoer, verwerking en afvoer wat vereis sal word.
- Moontlike bewyse wat ingehandig kan word: Vraelyste, notas (of opnames) wat gemaak is tydens onderhoude, fotos, dokumente/vorms wat in die huidige stelsel gebruik word, audio-/video-opnames van onderhoude/hoe die stelsel werk, ens.
- Inligting moet verskaf word omtrent die besonderhede van die onderhoude, vraelyste, gebruikers, ens. Inligting soos die tyd en datum wanneer die vraelys/onderhoud gehou is asook die name, die plek en kontakbesonderhede van die deelnemers moet voorsien word. Jou onderwyser moet hierdie inligting kan gebruik om te verifieer dat die ondersoek wel gedoen is soos wat aangedui is in die bewyse wat jy voorsien het.

## 4. Moontlike oplossing

- Gebruik jou navorsing en die terugvoer van die gebruikers en skryf 'n moontlike/beoogde oplossing vir die probleem in jou eie woorde.

- Moontlike bewyse wat ingehandig kan word: Definisie van die probleem en/of lys van vereistes. Dit is nie 'n getaileerde spesifikasie met toevoer en afvoer nie maar eerder slegs 'n eenvoudige lys van eienskappe wat die beoogde oplossing sal voorsien aan gebruikers.

#### 5. Voorsien 'n beplanningsdokument

Kombineer die resultate van jou navorsing in die vorm van 'n beplanningsdokument wat uit die volgende bestaan:

- Beskrywing van die probleem
- Agtergrond-inligting oor die onderwerp met verwysings
- Bewyse van die metodes wat gebruik is om inligting in te samel sowel inligting omtrent die gebruikers of die persone wat daaraan deelgeneem het
- Breë oorsig van 'n moontlike oplossing vir die probleem

#### 6. Handig 'n dokument in met die volgende:

- Beskrywing van die probleem
- Bewys van die metodes waarop inligting ingesamel is
- Programspesifikasies (Soos aangedui hierbo)
- Breë oorsig van die oplossing van die probleem

**FASE 2:****Ontwerp****Sperdatum:** \_\_\_\_\_

Met die voltooiing van hierdie fase moet jy die spesifikasies van die program/stelsel bepaal en aandui hoe die program/stelsel die doelwitte sal bereik wat tydens die analise daar gestel is. **Bestudeer die assesseringsinstrument vir Fase 2 en 3 om seker te maak dat jou projek voldoen aan die vereistes waaraan alle projekte moet voldoen.**

1. Skakel vereistes/eienskappe om in spesifikasies  
Spesifiseer die oplossing in terme van:
    - Toevoer
    - Verwerking
    - Afvoer (tabelle, grafieke, geldeenheid, eenhede van meting, ens ingesluit)
- Sluit die volgende in as deel van die spesifikasies:
- Defensiewe programmeringstegnieke soos:
    - Data validering (dui aan waar dit toegepas moet word en hoe)
    - Foutboodskappe
  - Datastrukture om data te organiseer en te stoor
    - Databasisontwerp
      - velde (tipes, name en groottes)
      - verwantskappe
      - sleutels (primêre en vreemde)
      - ER diagramme
    - Datatipes wat in programmering gebruik gaan word:
      - Primitiewe datatipes – enkele waardes, bv. integer en real/double
      - Gestruktureerde datatipes – versamelings van data, bv. skikkings, matrikse, rekords, stelle (sets), combo boxes, lyste, vektore
      - Abstrakte datatipes – versamelings van stelle data en stelle bewerkings wat op die data uitgevoer kan word, bv. klasse en objekte
      - Ontwerp van klasse – metodes met parameters en terugstuur datatipes, attribute, konstruktors, klas-diagramme, oorerwingshierargieë waar van toepassing (hoër orde)
  - Grafiese gebruikerskoppelvlak-ontwerp (GUI)  
Tekeninge en/of skermsafdrucke (“screen shots”) van die GUI wat die volgende aandui:
    - Uitleg van die komponente
    - Navigering (bv. hoe die gebruiker van een skerm na die volgende beweeg en terug)
    - Vloei van gebeure (van een skerm na die volgende, van een gebeurtenis na die volgende)

- Vloei van data tussen eenhede/modules (watter data word “gestuur” tussen modules?)
- Motivering vir die gebruik van toevoer- en afvoer-komponente (bv. hoekom is a “combo box” gebruik in plaas van ‘n “text field” of hoekom is ‘n “check box” gebruik in plaas van ‘n “radio button”)

2. Handig ‘n dokument in met al die programspesifikasies soos hierbo aangedui. Die dokument moet die volgende insluit:

- Toevoer, verwerking en afvoer vereistes
- Fout-toetsing en valideringsprosedures
- Databasisontwerp
- Lys van datastrukture wat gebruik gaan word (insluitend klasdiagramme)
- Ontwerp en uitleg van die GUI

Die dokument kan TVA-tabelle, vloei-diagramme and diagramme soos EV-diagramme, klasdiagramme en /of skermafdrukke insluit wat ‘n duidelike voorstelling van die stelsel gee met die nodige aantekeninge en beskrywings van spesifikasies.

**FASE 3:****Kodering en Implementering****Sperdatum:** \_\_\_\_\_

Met voltooiing van hierdie fase sal jy die oplossing moet kodeer met insluiting van die grafiese gebruikerskoppelvlak (GUI) soos wat dit beplan is in die vorige fase, skep van datastrukture en ontfouting en toetsing van die program. **Bestudeer die assesseringsinstrument vir Fase 3 om seker te maak dat jou projek voldoen aan die vereistes waaraan alle projekte moet voldoen.**

Voorgestelde stappe om te volg om hierdie fase te voltooi:

1. Breek die oplossing soos beskryf in fase 2 op in modules (bv. volgens opsies).
2. Kodeer/Skep die Grafiese Gebruikerskoppelvlak (GUI)
3. Skep die datastrukture om die data te organiseer en te stoor.
4. Kodering – Kodeer die programoplossing volgens die vereistes en die spesifikasies (toevoer, verwerking en afvoer) wat in die vorige fase opgestel is. Punte sal onder andere toegeken word vir:
  - Toevoer uit tekslêer
  - Toegang tot die databasis
  - Manipulering van stringe
  - Berekeninge
  - Goeie programmeringsbeginsels: hergebruik van kode, veranderlike name, kommentaar
  - Parameter-oordrag: Onafhanklikheid van modules
5. Hantering van foute: Maak seker dat toevoer gevalideer word en dat hantering van uitsonderings toegepas word waar nodig.
6. Toetsing /Ontfouting – Maak seker dat die program reg uitvoer en geen foute het nie
  - Gebruik 'n **toets-strategie** om te verseker dat die program/stelsel doen wat dit behoort te doen deur gebruik te maak van verskillende reekse data insluitend ekstreme en foutiewe data. Hou boek van hierdie data aangesien dit benodig sal word in jou dokumentasie.
7. Evalueer jou program/stelsel – Doen die program wat dit moet doen/voldoen dit aan die vereistes? Het jy goeie programmerings-beginsels toegepas?
  - Hersien die programkode. Is goeie programmeringsbeginsels toegepas? Bestudeer die assesseringsinstrument vir fase 3 deeglik.
  - Werk die program/stelsel reg?
  - Stem die program ooreen met wat in Fase 1 en 2 aangedui word
  - Is die Grafiese Gebruikerskoppelvlak (GUI) goed ontwerp

Nota:

Jy word toegelaat om gebruik te maak van “geleende” kode maar dit mag nie meer as 10% van jou programmeringskode uitmaak nie. Dit sal tipies kode wees om uitsonderlike/ongewone funksies uit te voer soos om 'n videouittreksel te speel ens. **Jou program sal nie deur die onderwyser aanvaar word indien “geleende” kode die perk van 10% oorskrei nie.**

**FASE 4:****Dokumentering & Evaluering****Sperdatum:** \_\_\_\_\_

Met voltooiing van hierdie fase met jy die dokumentasie finaliseer wat by die oplossing pas.

**1. Tegniese Dokumentasie**

Die tegniese dokumentasie/handleiding moet die volgende insluit:

- Algemene beskrywing van die probleem en die oorsigtelik oplossing van die probleem soos saamgestel in fase 1
- Databasisontwerp. Skermafdrucke van die “design view” van die velde en hulle datatipes van al die tabelle asook die verwantskappe tussen die tabelle of ‘n beskrywing daarvan.
- Drukstukke van die kode van ongewone stelsel funksies wat gebruik is byvoorbeeld die gebruik van die stelselklok/CDspeler.
- Spesifikasies van die toets-prosedures en die toets-resultate.
- Verwantskappe/ kommunikasie tussen modules (bv. Dui aan watter skerms/modules met mekaar kommunikeer en watter data tussen hulle oorgedra word)
- Voorbeeld van afvoerloopies van die program met resultate: ‘n Drukstuk van die resultate wat die tipiese data aantoon wat gebruik sal word. Die program moet ook getoets word met ekstreme en foutiewe data.
- Volledige drukstuk van die bronkode van al die modules

**2. Gebruikershandleiding**

Die gebruikershandleiding moet die volgende insluit:

- Titelbladsy en inhoudsopgawe.
- Agtergrond van die projek: Die geskiedenis van die ontwikkeling van die projek moet hier beskryf word.
- Hoe om die programmatuur te gebruik – detail hang af van die volledigheid van die programmatuur en die gebruikersvriendelikheid van die koppelvlak en die help-funksies.
- Omvang en beperkings van die program
- Apparatuur wat vereis word om die program uit te kan voer (hardeskyfspasie, geheue, SVE, spesiale toevoertoestelle, ens.)
- Programmatuur wat vereis word buiten die program self, byvoorbeeld voorafopgestelde klasse, bedryfstelsel, ens.
- Installeringsinstruksies.
- Lêers wat gebruik word en die inhoud daarvan. Die formaat/uitleg van elke lêer moet ingesluit wees.
- Gedetailleerde instruksies vir die gebruiker en ‘n deur-stap van die program.
- Toevoer wat vereis word. Die presiese formaat van die toevoer moet gespesifiseer word, veral as geformateerde toevoer gebruik word soos byvoorbeeld ‘n datum.
- Afvoer/verslae wat gelewer moet word.
- Toetsloopies van potensiële probleme
- Toekomstige ontwikkelings. Indien meer hulpbronne beskikbaar sou wees, watter addisionele funksies kon geïmplementeer gewees het?
- Verwysings en erkennings, veral waar programkode “geleen” is.

### 3. Handig in:

- Elektroniese kopie van programme en al die betrokke lêers, bv. datalêers, ens.
- Tegniese Handleiding
- Gebruikershandleiding

### 4. Demonstreer die program vir evaluering en assessering

#### Riglyne vir die demonstrasie van die projek:

- Die onderwyser sal datums en tye skeduleer vir demonstrasies - ongeveer 20 minute sal per projek toegelaat word.
- Die leerder moet al die dokumentasie ten minste 1 week voordat die demonstrasie plaasvind inhandig.
- Die demonstrasie word elektronies op 'n rekenaar gedoen.
- Die leerder moet sy/haar program op die rekenaar uitvoer en al die funksies wat die program bied aan die onderwyser uitwys vir evaluering.
- Die onderwyser kan die toetsstrategie wat in die tegniese handleiding voorkom gebruik as riglyn en die leerder vra om sekere of al die toetsprosedures uit te voer.
- Die onderwyser kan addisionele toetsprosedures laat uitvoer om seker te maak dat die hele program werk.
- Die onderwyser gebruik die nasienblad vir Fase 3 as riglyn en ken punte daarvolgens aan die projek toe tydens die demonstrasie.
- Die onderwyser moet **willekeurig 'n paar stukke programmeringskode (uitsluitend die 10% geleende kode) in die projek uitwys wat die leerder aan die onderwyser moet verduidelik**. Dit word gedoen om seker te maak dat alle kode deur die leerder self geskryf is. 'n Soortgelyke prosedure sal ook tydens eksterne moderering gevolg word. Indien die leerder nie op hoogte is van die kode wat in sy eie projek gebruik is nie, kan geen punte vir die projek aan die leerder toegeken word nie.
- Die leerder handig na afloop van die demonstrasie die elektroniese weergawe van die projek in sodat die onderwyser weer daarna kan kyk indien nodig voordat die finale punt toegeken word.

### 5. Finale algemene evaluering

Die onderwyser sal die volgende evalueer:

- Bestuur van tyd – het jy by al die sperdatums gehou?
- Waarde van die program – Is jou oplossing van toepassing binne die konteks van die gegewe scenario?
- Hou die verskillende fases van ontwikkeling verband met mekaar en lei dit tot die finale oplossing as een aaneenlopende proses?

# **Inligtingstechnologie**

## **Praktiese Assesseringtaak (PAT)**

### **2009**

## **Assesseringsinstrumente**

Hierdie afdeling bestaan uit 9 bladsye

# Assesseringsinstrumente vir die programmeringsprojek

Assessering vir Fase 1:

Naam van leerder:

Navorsing en Analise: Kriteria			Moontlike Punt	Punt Toegeken	
Probleemstelling	<b>Die probleem/ taak en doel daarvan word duidelik gestel en beskryf:</b>			3	
	3	2	1-0		
	Duidelik en ondubbelsinnig beskryf – dui duidelik aan wat die probleem/taak behels. Omskryf die aspekte wat opgelos moet word. Duidelike beskrywing wat die doel van die stelsel is.	Die beskrywing is vaag. Laat die leser onseker oor wat die doel van die stelsel sal wees.	Die beskrywing is so vaag dat geen doel vir die stelsel gevind kan word nie.		
Navorsing	<b>Die ondersoek dui duidelik aan/verduidelik sleutel areas van die onderwerp</b>			4	
	4	2-3	1-0		
	Omvattend. Toon goeie insig en verstaan al die sleutel areas van die onderwerp.	Minder omvattend – geringe tekortkominge. Toon insig in die meeste van die sleutel areas van die onderwerp.	Vaag. Min aanduiding dat die leerder enige van die sleutel areas verstaan word.		
	<b>Lys van verwysings ingesluit</b>			2	
2	1	0			
	Volledig. In aanvaarbare formaat	Minder volledig – geringe tekortkominge	Geen verwysings		
Vereistes	<b>Vereistes wat verkry is van die eind-gebruiker deur navorsing</b>			8	
	8-6	5-4	3-0		
	Volledige lys van vereistes wat goed omskryf is en verkry is van gebruikers deur van ten minste twee verskillende metodes gebruik te maak	Minder volledige lys van vereistes verkry van die gebruiker deur van ten minste een metode gebruik te maak. Nie so duidelik omskryf nie, maar steeds aanvaarbaar	Min of geen vereistes van die gebruiker verkry of vereistes opgemaak deur die leerder. Vaag en nie duidelik omskryf nie.		
	Bewyse van insameling van inligting met betrekking tot vereistes bv. vraelyste, voorbeelde van dokumente, notas/audio/video van onderhoude/navorsing ens. (1 punt vir elke relevante bewysstuk tot 'n maksimum van 4)			4	
Moontlike oplossing	<b>'n Moontlike oplossing is gegee:</b>			5	
	5	4-3	2-0		
	Duidelik en volledig beskryf – 'n duidelike oorsig van 'n moontlike oplossing is gegee. Die leser weet presies wat die stelsel sal doen.	Basiese beskrywing is gegee, maar sommige aspekte van die voorgestelde oplossing is vaag en laat die leser onseker oor sommige aspekte van die oplossing.	Die beskrywing is vaag. Geen duidelike oplossing kan geïdentifiseer word in die beskrywing nie.		
	<b>Die omvang van die voorgestelde oplossing is goed gedefinieer (Eienskappe of opsies wat ingesluit sal word)</b>			1	
	<b>Geskiktheid van die voorgestelde oplossing in die konteks van die scenario:</b>			3	
3	2	1-0			
	Uiters geskik. Goeie toepassing in die konteks van die scenario	Geskik. Die toepassing in die konteks van die scenario is nie altyd oortuigend nie	Baie min of glad nie geskik. Toepassing hou nie verband met die gegewe scenario nie		
<b>Totaal:</b>			<b>30</b>		

## Assessering vir Fase 2:

Naam van leerder:

Ontwerpfase: Kriteria				Moontlike Punt	Punt Toegeken	
Spesifikasies	<b>Toevoer: Beskrywing van die toevoer</b>				3	
	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>		
	Alle toevoer duidelik beskryf in terme van die gebruiker se toevoer en ander bronne van toevoer.	Meeste van die toevoer is beskryf in terme van gebruiker-toevoer en ander bronne van toevoer.	Die beskrywing van die toevoer is vaag en onvolledig.	Slegs genoem – geen beskrywing		
	<b>Formaat van die toevoer (bv. Datum formaat jj/mm/dd) duidelik beskryf waar nodig</b>				2	
	<b>2</b>	<b>1</b>	<b>0</b>			
	Formaat van al die toevoer is duidelik beskryf waar nodig.	Beskrywings van die formaat van die toevoer is vaag en onvolledig. Nie altyd beskryf nie	Formaat van toevoer is nie beskryf nie.			
	<b>Verwerking: Dui aan waar en hoe verwerking of manipulering van data gedoen moet word om sekere afvoer te verkry (bv. die formaat van die navraag of hoe die koste van items bereken moet word, ens.)</b>				5	
	<b>5</b>	<b>4-3</b>	<b>2-1</b>	<b>0</b>		
	Kort, duidelik en duidelike beskrywing van verwerking/manipulering van data in alle gevalle waar van toepassing. Gebruiker altyd seker van die resultaat.	Beskrywing van verwerking/manipulering van data gedoen/duidelik in die meeste gevalle waar van toepassing. Goeie poging maar kan verbeter. Nie altyd korrek of toepaslik	Beskrywing van verwerking/manipulasie van data nie duidelik in meeste gevalle nie. Nie altyd gedoen nie. Beskrywings vaag/onvolledig. Nie korrek/toepaslik nie.	Verwerking/manipulasie van data is nie beskryf nie.		
	<b>Afvoer</b>				3	
	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>		
	Het al die nodige afvoer ge-identifiseer volgens die spesifikasies.	Het die meeste van die nodige afvoer ge-identifiseer volgens die spesifikasies.	Het minder as 50% van die afvoer ge-identifiseer volgens die spesifikasies.	Het nie die afvoer geidentifiseer nie.		
	<b>Formaat van die afvoer (bv. geldeenheid, tabel-formaat, eenhede waarin gemeet, ens.) duidelik beskryf waar nodig</b>				2	
	<b>2</b>	<b>1</b>	<b>0</b>			
	Formaat van afvoer duidelik gedefinieer in alle gevalle. Voldoen aan die vereistes.	Die beskrywing van formaat van afvoer is vaag of onvolledig	Formaat van die afvoer is nie gedefinieer nie			
<b>Datavalidering</b>				2		
<b>2</b>	<b>1</b>	<b>0</b>				
Aangedui vir alle toepaslike toevoer en beskryf in detail	Aangedui in sommige gevalle waar van toepassing of validering is vaag en onvolledig	Nie aangedui nie				
<b>Foutboodskappe wat met datavalidering geassosieer word</b>				2		
<b>2</b>	<b>1</b>	<b>0</b>				
Aangedui vir alle toepaslike valideringstoetse/foute en in detail beskryf/vertoon	Sommige foutboodskappe aangedui. Beskrywing van die boodskappe vaag	Geen foutboodskappe aangedui vir enige van die validerings nie.				

Ontwerp	<b>Oorsig van die ontwerp van die databasis in terme van tabelle en velde:</b>				8								
	2		1				0						
	Velde, veldtipes en groottes is goed gekies en geskik vir die toepassing.		Meeste velde, veldtipes en groottes is goed gekies vir die toepassing.				Velde, veldtipes en groottes nie goed gekies vir die toepassing nie.						
	2		1				0						
	Veld-eienskappe soos validering en maskers is aangedui waar van toepassing.		Meeste veld-eienskappe soos validering en maskers is aangedui waar van toepassing.				Veld-eienskappe soos validering en maskers nie aangedui nie.						
	4		3				2-1		0				
	Velde goed gegroepeer in tabelle. Geen onnodige herhaling van velde in tabelle. Meer as twee tabelle met verwantskappe tussen die tabelle. Dui regte primêre en vreemde sleutels aan.		Velde meestal goed gegroepeer in tabelle. Soms herhaling van velde in tabelle. Ten minste twee tabelle met regte verwantskap tussen die tabelle. Dui regte primêre en vreemde sleutels aan.				Velde meestal nie goed gegroepeer in tabelle/heelwat herhaling van velde in tabelle. Ten minste twee tabelle sonder/met verkeerde verwantskap tussen die tabelle. Geen/foutiewe primêre en vreemde sleutels aangedui.		Slegs een tabel of slegs velde aangedui				
	<b>Abstrakte datastrukture om data te organiseer en te stoor</b> (versamelings met stelle data en stelle bewerkings wat op die data uitgevoer kan word bv. klassen en objekte)						5						
	5		4-3						2-1		0		
	Gebruik abstrakte datatipes. Toepassing is goed beplan in terme van klasse en objekte. Objekte goed saamgestel met relevante metodes.		Gebruik abstrakte datatipes. Ruimte vir verbetering. Toepassing is nie heeltemal beplan in terme van klasse en objekte nie. Nie altyd toepaslik nie. Die hele program is nie objek-georiënteerde nie.						'n Poging is aangewend om abstrakte datatipes te gebruik. Objekte nie altyd sinvol saamgestel nie./objekte is saamgestel maar nie reg/enigsins gebruik nie. Baie min van die program is objek-georiënteerd.		Geen abstrakte datatipes (klasse en objekte)		
<b>GUI ontwerp: Geskikte toevoer-komponente om akkurate/geldige toevoer van data te ondersteun</b>				3									
3		2				1			0				
Geskikte toevoer-komponente in alle gevalle waar van toepassing. Motiveer keuses van komponente waar nodig.		Geskikte toevoer-komponente in meeste van die gevalle waar van toepassing. Keuse nie altyd gemotiveer nie.				Geskikte toevoer-komponente in minder as 50% van die gevalle waar van toepassing.			Nie gedoen nie/nie geskikte komponente in alle gevalle.				
<b>GUI-ontwerp: Afvoer</b>						3							
3		2								1		0	
Het geskikte afvoer-komponente geïdentifiseer in alle gevalle waar van toepassing.		Het geskikte afvoer-komponente geïdentifiseer in die meeste van die gevalle waar van toepassing.								Het geskikte afvoer-komponente geïdentifiseer in minder as 50% van die gevalle waar van toepassing.		Nie gedoen nie/nie geskikte afvoer-komponente in al die gevalle nie.	
Algemeen	<b>Spreek die vereistes aan wat in fase 1 gespesifiseer is</b>									2			
	2		1				0						
Al die vereistes aangespreek		Omtrent 50%					Minder as 50%						
<b>Totaal:</b>							<b>40</b>						

Assessering vir Fase 3:

Naam van leerder: \_\_\_\_\_

**NB: Fase 3 en 4 sal slegs geassesseer sodra fases 1 en 2 voltooi is EN indien fase 1 en 2 van toepassing is op die onderwerp waarvoor fase 3 en 4 ontwerp is**

Kodering & Implementeringsfase: Kriteria			Moontlike Punt	Punt Toegeken	
Databasis	<b>GEEN DATABASIS = 0 vir al die databasis-afdelings</b>			3	
	<b>Databasis is toepaslik en effektief geskep</b>				
	<b>3</b>	<b>2-1</b>	<b>0</b>		
	Aantal tabelle is relevant tot 'n effektiewe oplossing – tabelle is genormaliseer. (Gesките hoeveelheid tabelle om die oplossing effektief te ondersteun)	Aantal tabelle dui 'n gedeeltelike oplossing aan – tabelle nie genormaliseer nie (Kan meer tabelle hê vir 'n meer effektiewe oplossing)	Aantal tabelle irrelevant tot die oplossing. (Slegs een tabel of 'n aantal onverwante tabelle met herhaling van velde. Of meer as een databasis met een tabel in elke databasis in plaas van een databasis)		
	<b>Primêre en vreemde sleutels: (Geen primêre sleutels = 0)</b>				
	<b>3</b>	<b>2</b>	<b>1</b>		
	Primêre en vreemde sleutels in verwante tabelle effektief en geskik. Al die verwantskappe is goed saamgestel met die regte primêre en vreemde sleutels	Ten minste een effektiewe en geskikte verwantskap met die regte primêre en vreemde sleutels. Sommige primêre sleutels of vreemde sleutel in die verwante tabel is nie korrek nie.	Meeste primêre sleutels of vreemde sleutels is nie geskik nie of slegs die verstek ( <i>default</i> ) sleutels is gebruik.		
	<b>Veldtipes – databasis: (Slegs verstek (<i>default</i>) datatipes gebruik = 0)</b>				
	<b>2</b>	<b>1</b>			
	Al die velde met geskikte datatipes.	Datatipes van sommige van die velde is nie geskik nie.			
<b>Veldgroottes – databasis: (Slegs verstek groottes gebruik = 0)</b>					
<b>2</b>	<b>1</b>				
Al die velde met geskikte groottes vir die toepassing.	Sommige velde te groot of te klein.				
Beskrywende name vir die velde in die databasis			1		
<b>NOTA: Evalueer die kode indien die program nie uitvoer nie. Hierdie afdeling het te doen met programmering in Delphi/Java uitsluitend SQL</b>					
Programmeringsoplossing	<b>Datastrukture word effektief en reg gebruik uitsluitend databasis tabelle</b>			16	
	<ul style="list-style-type: none"> <li>Primitiewe datatipes (enkele waardes bv. integer, real, ens.) 1 punt per <b>verskillende</b> geskikte datatipe tot by 'n maksimum van 4 punte</li> <li>Gestruktureerde datatipes (versamelings van data bv. skikkings, matrikse, versamelings, records, combo boxes, lyste, vektore) 2 punte vir elke <b>verskillende</b> tipe datastruktuur tot by 'n maksimum van 4 punte</li> <li>Abstrakte datatipes (versameling van stelle data met bypassende stelle metodes wat die relevante data kan manipuleer bv. klasse wat objekte en metodes definieer) 4 punte vir elke definiering van so 'n versameling of toepaslike gebruik daarvan tot by 'n maksimum van 8 punte</li> </ul>				
	<b>Veranderlike name/names van datastrukture is sinvol</b>				
<b>Seleksie en herhaling</b>			1		
Effektiewe en geskikte gebruik van beheerstrukture regdeur die program. 1 punt elk bv. case/switch wanneer meer effektief is as genestede if, gebruik van vlag wanneer geskik, ens) Maksimum van 4 punte			4		

<b>Komplekse programmeringstegnieke wat goed gebruik word soos die speel van stukkie video, geleende kode, threads, netwerkprosesse, ens.</b>					
<b>6</b>	<b>5-3</b>	<b>2-1</b>	<b>0</b>		
Werk reg. Reg gebruik en voeg waarde toe aan die oplossing	Werk reg. Nie altyd reg gebruik nie en voeg nie regtig waarde by tot die oplossing nie.	'n Poging is aangewend maar dit werk nie reg nie. Nie reg gebruik nie. Hou nie verband met die oplossing.	Geen komplekse tegnieke gebruik nie of dit is meer as 10% van die kode	<b>6</b>	
<b>Vlak van kompleksiteit van bewerkings/verwerkings regdeur die program. (Geen bewerkings = 0 punte)</b>					
<b>3</b>	<b>2</b>	<b>1</b>		<b>3</b>	
Gevorderd – graad 12 vlak en hoër	Middelvlak – Graad 11 vlak	Eenvoudig – graad 10 vlak			
<b>Stringmanipulasie reg gebruik. (Geen stringmanipulasie = 0)</b>					
<b>4</b>	<b>3-2</b>	<b>1</b>			
Geskikte stringhantering, goed toegepas, variasie van stringmanipulasie-instruksies effektief gebruik	Beperkte variasie van stringmanipulasie-instruksies is gebruik. Nie altyd die beste toepassing nie.	Uiters beperkte stringmanipulasie is gedoen. Baie eenvoudige toepassings. Meeste van die tyd nie reg gebruik nie.		<b>4</b>	
<b>Tekslêers gebruik. (Geen tekslêers gebruik = 0 punte)</b>					
<b>4</b>	<b>3-2</b>	<b>1</b>			
Effektief gebruik vir toevoer en/of afvoer van inligting. Die inhoud van die lêer(s) is geskik vir die toepassing. Lees /skryf reg na tekslêers	Effektief gebruik in die meeste van die gevalle. Die inhoud van die lêer(s) is geskik vir die toepassing. Geringe foute met die lees /skryf na tekslêers	Slegs 'n byvoeging. Nie goed toegepas/ of 'n poging is aangewend maar die kode werk nie.		<b>4</b>	
<b>Vloei van data en prosesse (oordrag van parameters)</b>					
<b>4</b>	<b>3-2</b>	<b>1</b>	<b>0</b>		
Uitstekende interaksie/kommunikasie tussen die modules/klasse. Sluit gevorderde datatipes in as parameters of terugstuurtypes	Meer as voldoende kommunikasie tussen modules/klasse met geringe foute. Sluit parameteroordrag tussen modules in.	Beperkte kommunikasie tussen modules/klasse. Slegs primitiewe datatipes word in baie beperkte mate oorgestuurd as parameters.	Geen kommunikasie tussen modules/klasse. Geen parameteroordrag	<b>4</b>	
<b>Hergebruik van kode (skep van klasse en metodes/ subprogramme)</b>					
<b>4</b>	<b>3-2</b>	<b>1</b>	<b>0</b>		
Geskikte en effektiewe hergebruik van kode and/of metodes. Modules is onafhanklik. Lokale veranderlikes word gebruik waar van toepassing. Globale veranderlikes slegs gebruik waar nodig.	Daar is hergebruik van kode en /of metodes maar dit is nie altyd effektief/ geskik nie Modules nie altyd onafhanklik nie. Sommige globale veranderlikes beperk onafhanklikheid van modules.	Voldoende Daar is hergebruik van kode. Sommige metodes kon opgebreek gewees het in meer modules. Omtrent al die veranderlikes is globaal verklaar.	Linier programmering – net een aaneenlopende program, geen modules. Geen hergebruik van kode en/of metodes.	<b>4</b>	

	<b>Algoritmes van die oplossings</b>				
	<b>5</b>	<b>4-3</b>	<b>2-1</b>		
	Al die algoritmes wat gebruik is om die probleem op te los is geskik en effektief. Geen logiese foute. Die resultate van al die verwerkings is korrek.	Die meeste van die algoritmes wat gebruik is om die probleem op te los is geskik en effektief. Min logiese foute. Sommige resultate van verwerkings is foutief	Minder as 50% van die algoritmes wat gebruik is om die probleem op te los is geskik en effektief. Baie logiese foute. Feitlike al die resultate is foutief/lewer min van die verwagte resultate		
	<b>Afvoer teenoor vereistes. (Geen afvoer = 0 punte)</b>				
	<b>3</b>	<b>2</b>	<b>1</b>		
	Afvoer voldoen aan al die vereistes van die oplossing.	Afvoer voldoen in die meeste gevalle aan die vereistes van die oplossing.	Afvoer voldoen aan minder as 50% van die vereistes van die oplossing.		
	<b>Struktuur van die afvoer (kode). Geen kode om afvoer te vertoon = 0 punte</b>				
	<b>3</b>	<b>2</b>	<b>1</b>		
	Afvoer is altyd goed gestruktureerd, leesbaar met opskrifte en onderopskrifte. Opskrifte word herhaal op die volgende bladsy/skerm waar van toepassing.	Afvoer is in die meeste van die gevalle goed gestruktureerd, leesbaar met opskrifte en onderopskrifte. Opskrifte word in die meeste van die gevalle herhaal op die volgende bladsy/skerm waar van toepassing.	Afvoer is nie goed gestruktureerd, leesbaar met opskrifte en onderopskrifte. Opskrifte is in die meeste van die gevalle nie goed geformuleer nie of afwesig. Word meestal nie op die volgende bladsy/skerm waar van toepassing herhaal nie.		
	<b>Formaat van afvoer: alle waardes is reg geformateer waar van toepassing, bv. geldeenheid, eenheid waarin gemeet, ens.</b>				
<b>Databasis interaksie</b>	<b>Databasis se konneksie-string/pad is reg opgestel en dit werk</b>			<b>2</b>	
	<b>Databasis is interaktief met die program op 'n betekenisvolle manier soos navrae en verslae</b>			<b>2</b>	
	<b>Manipuleer rekords reg dmv SQL (2 punte vir elke korrekte SQL-stelling tot by 'n maksimum van 8 punte)</b>			<b>8</b>	
	Insert, delete, select, update. Noem ander:				
	<b>Manipuleer velde dmv SQL (2 punte vir elke korrekte SQL-stelling tot by 'n maksimum van 6 punte)</b>			<b>6</b>	
	Berekeninge met velde, verander die inhoud, vertoon slegs sekere velde, gee name vir velde, sorteer volgens velde. Noem ander:				
<b>Fout-hantering en toetsing</b>	<b>Program kan suksesvol kompilleer – geen sintaksfoute</b>			<b>2</b>	
	<b>Uitvoerfoute (0 = geen van die opsies suksesvol uitgevoer kan word nie)</b>			<b>4</b>	
	<b>4</b>	<b>3-2</b>	<b>1</b>		
	Geen uitvoerfoute. Al die opsies voer suksesvol uit.	Sommige opsies veroorsaak foute wanneer dit uitgevoer word.	Slegs een of twee opsies kan suksesvol uitgevoer word.		
	<b>Toevoer word gevalideer dmv kode en/ of eienskappe van komponente (Geen = 0)</b>			<b>3</b>	
<b>3</b>	<b>2</b>	<b>1</b>			
Alle toevoer wat gevalideer moet word, word gevalideer.	Die meeste van die toevoer wat gevalideer moet word, word gevalideer.	Minder as 50% van die toevoer wat gevalideer moet word, word gevalideer.			

	<b>Geskikte foutboodskappe word vertoon (Geen foutboodskappe = 0)</b>			3		
	3	2	1			
	Geskikte en gebruikersvriendelike foutboodskappe in alle gevalle waar data validering toegepas word.	Geskikte en gebruikersvriendelike foutboodskappe in die meeste van die gevalle waar data validering toegepas word.	Geskikte en gebruikersvriendelike foutboodskappe in 'n paar van die gevalle waar data validering toegepas word			
	<b>Program lewer afvoer (Afvoer is die resultaat van verwerking, i.e. GUI vensters met geen funksionaliteit klassifiseer nie as afvoer nie)</b>					2
	<b>Program gee regte afvoer/wat verwag word (Geen afvoer/slegs verkeerde afvoer = 0)</b>					3
	3	2	1			
	Program gee regte en geskikte afvoer in alle gevalle.	Program gee regte en geskikte afvoer in die meeste gevalle.	Program gee regte en geskikte afvoer in minder as 50% van die gevalle.			
	<b>Hantering van uitsonderinge (Exception handling)</b>					2
2	1	0				
Gebruik try...catch tegniek om uitsonderings te hanteer	'n Poging is aangewend om uitsonderings te hanteer of gebruik van if-stellings om foute te hanteer	Geen poging				
Grafiese Gebruikerskoppelvlak (GUI)	<b>Verskillende skerms is effektief gebruik (windows/panels/tabsheets, ens)</b>			1		
	<b>Komponente gebruik vir toevoer/afvoer (geïmplementeer soos beplan in fase 2 OF nodige veranderinge is gemaak volgens terugvoer van fase 2)</b>			2		
	2	1				
	Al die toevoer-komponente op al die skerms is geskik vir die toepassing soos aangedui in fase 2 of die nodige wysings is gemaak volgens terugvoer na evaluering van fase 2	Die meeste van die toevoer-komponente is geskik soos aangedui in fase 2 of die nodige wysings is gemaak volgens terugvoer na evaluering van fase 2				
	<b>Byskrifte/lokboodskappe met die presiese formaat van toevoer</b>			2		
	2	1				
	Konstant deur die hele projek toegepas waar nodig.	Toegepas in sommige gevalle waar nodig.				
	<b>Deurlopend dieselfde uitleg (dieselfde "look and feel" regdeur die program)</b>			2		
	2	1				
	Vir al die skerms (dieselfde kleure, skrifstyle regdeur die program gebruik)	Meeste van die skerms (verskillende kleure, skrifstyle ens. by sommige skerms)				
	<b>Afvoer/ Verslae verskaf geskikte inligting en is maklik om te lees (Geskikte skrif tipe, grootte, uitleg, kleure)</b>			2		
	2	1				
	Vir al die afvoer op al die skerms	Vir meeste van die afvoer op die skerms				
	<b>Tipe toevoer/afvoer is saam gegroepeer (bv. die adres se inligting)</b>			2		
	2	1				
	Vir al die skerms	Vir meeste van die skerms				
<b>Maklik om tussen skerms te beweeg (navigeer)</b>			2			
<b>Hulp beskikbaar as deel van program</b>			2			
<b>Konteks-sensitiewe help beskikbaar (tool tip teks ingesluit)</b>			2			
<b>Ontwerp neem die tipe gebruiker in ag (ouderdomsvlak van geletteredheid, visueel gestremdheid, geskiktheid van grafika, ens.)</b>			2			
Algemeen	<b>Kommentaar by kode/aantekeninge om te verduidelik/beskrywings om program makliker te verstaan regdeur die program</b>			2		
	<b>Afdelings in die kode van die program duidelik geskei om leesbaarheid te bevorder (spasiering, kommentaar aan die begin van elke metode/subafdeling, ens.)</b>			2		
<b>Totaal:</b>			<b>130</b>			

## Assessering van Fase 4:

Naam van leerder:

Dokumentasie en Evaluering: Kriteria			Moontlike Punt	Punt toegeken	
Tegniese dokumentasie	Databasis ontwerp duidelik aangedui en verduidelik		1		
	Beskrywing van datastrukture wat gebruik is		1		
	Beskryf/dui die verwantskap tussen modules/programme aan		1		
	Beskrywing van ongewone kode – funksies, berekeninge, geleende kode, ens.		1		
	Spesifikasie van toetsprosedures en toetsresultate		2		
	Hoe om foute reg te maak en opspoor van foute ( <i>troubleshooting</i> ) duidelik beskryf		2		
	Voorbeelde van uitvoere met resultate		2		
	Drukstuk van bronkode		1		
Lys van lêers waaruit die program bestaan.		1			
Gebruikershandleiding	Titel blad		1		
	Inhoudsopgawe		1		
	Agtergrond van die projek (persoonlike motivering en keuse)		2		
	Inleiding tot die projek (genre)		1		
	Projek se omvang en beperkinge is duidelik beskryf		1		
	Vereistes van toevoer/formate		1		
	Afvoer/verslae wat geskep is (afdrukke van skerms ingesluit)		1		
	Gedetailleerde instruksies en deurstap (0 = nie gedoen nie)				
		3	2	1	
	Al die stappe wat gevolg moet word wanneer die program uitgevoer word is duidelik aangedui en ondersteun met skermskote waar nodig. Goed gestruktureerd met logiese vloei. Goed geformuleer – maklik om te verstaan.	Die meeste van die stappe wat gevolg moet word wanneer die program uitgevoer word is duidelik aangedui en sommige skermafdrucke is gedoen. Nie altyd goed gestruktureerd met 'n logiese redelike vloei. Nie altyd goed geformuleer nie.	Baie min van die stappe wat gevolg moet word wanneer die program uitgevoer word is duidelik aangedui. Nie goed gestruktureerd. Nie goed geformuleer nie. Geen afdrukke van skerms.	3	
	Hardware wat vereis word om die program uit te voer. (1 Punt elk – maks 3 punte)		3		
	Spasie op die hardeskyf, geheue, SVE (CPU), spesiale toestelle – Gee lysLys:				
	Lys van programmatuur wat vereis word.		2		
	Installering sinstruksies is gegee		1		
Lys van lêers wat vereis word en die uitleg van die lêers		1			
Hantering van foute wat gebruiker kan maak duidelik omskryf.		2			
Toekomstige ontwikkelinge/uitbreidings word beskryf		2			
Verwysing en erkenning van bronne van geleende kode		1			
Algemene Evaluering	Bestuur van tyd: (0 = altyd laat, werk was nooit gedoen nie)				
	5	4-3	2-1		
	Al die sperdatums is nagekom – al 4 fases en al die vereistes is gedoen vir elke fase.	Het 3 uit 4 sperdatums nagekom of betyds ingegee maar al die werk was nie gedoen nie.	Slegs twee sperdatums is nagekom of het betyds ingehandig maar meeste van die werk was nie gedoen nie.	5	
	Vermoë om die doel van die werk of kode wat willekeurig uitgewys is te verduidelik (0 = nie instaat om enigiets te verduidelik nie)				
	5	4-3	2-1		
Verduidelik al die uitgewysde kode duidelik en met selfvertroue. Toon uitstekende insig	Verduidelik uitgewysde kode met geringe tekortkominge. Toon redelike insig	Nie instaat om meeste van die uitgewysde kode te verduidelik nie. Gebrek aan insig.	5		
Gesiktheid van die oplossing binne die konteks van die scenario					
	5	4-3	2-1		
Die oplossing is 'n werkende oplossing wat toegepas kan word in 'n werklike situasie	Die oplossing is 'n werkende oplossing wat met 'n bietjie verandering toegepas kan word in 'n werklike situasie	Sommige dele werk. Sommige dele kan toegepas word in 'n werklike situasie	5		
<b>Totaal:</b>			<b>50</b>		