



education

Department:
Education
REPUBLIC OF SOUTH AFRICA

INLIGTINGSTEGNOLOGIE

RIGLYNE VIR PRAKTIESE ASSESSERINGTAAK

2010

Hierdie riglyne bestaan uit 29 bladsye.

Inligtingstechnologie

Praktiese Assesseringstaak (PAT)

2009

Riglyne aan die onderwyser

A Inleiding

Die doel van die Praktiese Assesseringstaak 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 (inhoud, konsepte en vaardighede) in Inligtingstegnologie.

In Inligtingstegnologie tel die PAT 25% van die totale promosie-/sertifikaatpunt vir die vak (dit is 100 uit die 400 punte). Die PAT word geïmplimenteer oor die eerste drie kwartaal 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:

- Ondersoek en Analise
- Ontwerp
- Kodering en Implementering
- Dokumentasie en Algemene Evaluering

B Programmeringsprojek

1. Waaroor gaan die programmeringsprojek?

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

2. Beplanning van die programmeringsprojek

2.1 Beskrywing

Scenario vir die 2010 graad 12 PAT:

Die Department van Basiese Onderwys het 'n kompetisie van stapel gestuur vir al die graad 12 IT-leerders waarin hulle opvoedkundige programmatuur (programmatuur om leer en onderrig te fasiliteer) moet ontwikkel wat gebruik kan word om 'n konsep of vaardigheid in Suid Afrikaanse skole te onderrig en te leer.

Vir die doel van hierdie kompetisie moet die programmatuur gemik wees op enige konsep of inhoud van enige van die agt leerareas van graad R – 9 of enige van die 29 erkende vakke vir graad 10 – 12.

Jy mag jou eie onderwerp of fokusarea binne die konteks van die gegewe scenario kies.

Voorbeelde van moontlike tipe opvoedkundige programme om van te kies is:

- Dril- en praktiese oefeninge soos vir spelling of wiskundige konsepte
- Simulering van 'n wetenskapexperiment
- Toetsing van kennis van inhoud

Die leerder moet 'n onderwerp in een van die leerareas of vakke identifiseer en 'n program-oplossing ontwikkel wat onderwysers kan gebruik om sekere konsepte of inhoud te onderrig of wat leerders kan gebruik om sekere inhoud of konsepte te leer, te oefen of te dril. Die programmatuur kan ook wedywing, gebaseer op kompetisies tussen leerders of skole, aanmoedig.

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 leerder mag sy/haar eie onderwerp/toepassing binne die konteks van die gegewe scenario kies.

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

- 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 assesseringsinstrumente sal projekte penaliseer wat nie verwant is aan die gegewe scenario nie.

2.2 Ontwikkelingsfases van die projek

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

Fases	Punte	%
Fase 1: Ondersoek en Analise	33	16.5
Fase 2: Ontwerp	30	15
Fase 3: Kodering en Implementering	100	50
Fase 4: Dokumentasie en Algemene Evaluering	37	17.5
Totaal	200	100

Dokumentasie/bewys van wat die leerder gedurende elke fase gedoen het, moet met gespesifiseerde intervale 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 inlewering van die finale produk van elke fase sal deur die onderwyser vasgestel word. Die produk van elke fase sal geassesseer word en die punte sal aangeteken word.

2.3 Vereistes van die projek

Die leerder moet aandag gee aan die volgende minimum kriteria:

Ondersoek en Analise

- 'n Beskrywing van die probleem in die leerder se eie woorde waarin die hoofaspekte uitgelig word
- Ondersoek van die onderwerp: Die ondersoek 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, sowel as Fase 1 Asseseringsinstrument Fase 1)

Ontwerp

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

(Sien Leerderafdeling 2 Fase 2, Asseseringsinstrument Fase 2)

Kodering en Implementering

- Die projek moet die belangrikste ontwikkelingsinstrumente, naamlik databasisontwerp en programmering op 'n geïntegreerde manier insluit (Ander toepassings kan saam met hierdie ontwikkelingsinstrumente geïntegreer word).

- Ander aspekte van die programmeringsprojek wat geassesseer sal word:
 - Programmeringstyl
 - Grafiese Gebruikerskoppelvlak (GUI)
 - Gebruik van Mens-Rekenaar-Interaksiebeginsels
 - Vlak van kundigheid wat vereis word en funksionaliteit van die program
 - Robuustheid van die program, insluitend die gebruik van defensiewe programmeringstegnieke
 - Stem die projek ooreen met die oorspronklike mikpunte en doelwitte?

(Sien Leerderafdeling 2 Fase 3, Assesseringsinstrument Fase 3)

Dokumentering en Algemene Evaluering

- Dokumenteer die oplossing, installeringsprosedures asook apparatuur- en programmatuurvereistes – Tegnieiese Handleiding
- Stel 'n gebruikershandleiding saam.
- Evalueer die volgende:
 - Tydsbestuur van die leerder – Het hy/sy al die take betyds ingelewer?
 - 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 voltooi word en deur die onderwyser gefasiliteer en deurlopend gemonitor word.

Sien Assesseringsinstrumentafdeling vir die assesseringsblaaie vir die verskillende fases.

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

- Die "Instruksies aan die Leerders"-afdeling wat in hierdie dokument ingesluit is
- Die assesseringsblaaie 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 vir 'n ondervragingsessie aan die einde van fase 3 te demonstreer. Onderwysers moet die projekte volgens die assesseringsinstrument wat voorsien is vir fase 3, assesseer.

Leerders sal **NIE** toegelaat word om die onderwerpe van hulle projekte te verander nadat fase 1 en 2 voltooi en geassesseer 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 aangegeven word. Fase 3 sal nie geassesseer word nie tensy fase 1 en 2 navorsing weerspieël wat vir die nuwe onderwerp gedoen is nie.

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

Riglyne vir die demonstrasie en interne assessering:

- Die onderwyser moet datums en tye vir demonstrasies skeduleer. Laat ongeveer 15 minute toe per projek asook 5 minute vir die opstelling en opsommende terugvoering 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, sodra 'n spesifieke fase van ontwikkeling geassesseer word. Die onderwyser moet die vordering tydens die projek in verband met die werk wat in vorige fases gedoen is, deeglik monitor.
 - Die vereistes wat geïdentifiseer is en uitstippel is in fase 1 moet gereflekteer word in fase 2 – die ontwerpfase.
 - Die werk van 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 tydens die demonstrasie toe te ken.
- Die onderwyser moet **willekeurig 'n paar gedeeltes 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 lewer 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.

Let wel: Nadat die produk van 'n fase ingelewer is en geassesseer is, sal dit nie weer geassesseer word nie.

6. Rekordhouding en Verslagdoening

Vir elke fase sal die onderwyser die fase assessee, die punt aanteken en terugvoering 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 programmeringsprojek bestuur kan word:

Opsie 1:

- Spandeer een of twee periodes per week aan die programmeringsprojek 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, kan die onderwyser teen die einde van die eerste kwartaal met die PAT begin en een fase per kwartaal voltooi.

Opsie 2:

- Die onderwyser kan verkies om 'n aaneenlopende tydperk aan die programmeringsprojek te spandeer, bv. die laaste week van elke kwartaal en een fase per kwartaal voltooi.

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

8. WENKE

- Verduidelik eers die PAT en gee 'n oorsig van die proses van ontwikkeling aan die leerders voordat hulle met Fase 1 begin.
- Bespreek die taak en die onderwerp met die leerders. Laat die leerders toe om vrae te vra en maak seker dat hulle baie goed verstaan wat die probleem is wat opgelos moet word.
- Bespreek voorbeelde van moontlike onderwerpe binne die gegewe scenario met die leerders. Laat die leerders self met idees vorendag kom en bespreek die geskiktheid daarvan.
- Alhoewel die scenario's verskillend sal wees kan dit 'n goeie idee wees om vir die leerders goeie en swak voorbeelde van PATé van vorige jare te wys.
- Wees baie streng met die inlewering van die fases.

INLIGTINGSTEGNOLOGIE
PRAKTIESE ASSESSERINGSTAAK (PAT)
2010

INSTRUKSIES AAN DIE LEERDER

INLIGTINGSTEGNOLOGIE – PRAKTIESE ASSESSERINGSTAAK (PAT)

Die IT PAT bestaan uit vier komponente:

- Ondersoek en Analise
- Ontwerp
- Implementering en Kodering
- Dokumentasie en Algemene Evaluering

1. Beplanning van die programmeringsprojek

1.1 Beskrywing

Scenario vir die 2010 Graad 12 PAT:

Die Departement van Basiese Onderwys het 'n kompetisie van stapel gestuur vir al die graad 12-IT leerders waarin hulle opvoedkundige programmatuur (programmatuur om leer en onderrig te fasiliteer) moet ontwikkel wat gebruik kan word om 'n konsep of vaardigheid in Suid Afrikaanse skole te onderrig of te leer.

Vir die doel van hierdie kompetisie moet die programmatuur gemik wees op enige konsep of inhoud van enige van die agt leerareas van graad R – 9 of enige van die 29 erkende vakke vir graad 10 – 12.

Jy mag jou eie onderwerp of fokusarea binne die konteks van die gegewe scenario kies.

Voorbeelde van moontlike tipe opvoedkundige programme om van te kies, is:

- Dril- en praktiese oefeninge soos vir spelling of wiskundige konsepte
- Simulering van 'n wetenskapeksperiment
- Toetsing van kennis van inhoud

Identifiseer 'n onderwerp in een van die leerareas of vakke en ontwikkel 'n program-oplossing wat onderwysers kan gebruik om sekere konsepte of inhoud te onderrig of wat leerders kan gebruik om sekere inhoud of konsepte te leer, te oefen of te dril. Die programmatuur kan ook wedywering, gebaseer op kompetisies, tussen leerders of skole aanmoedig.

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. Buiten 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 kan toepas:

- 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 assesseringsinstrumente sal projekte penaliseer wat nie verwant is aan die gegewe scenario nie.

Let wel: 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 en geassesseer word.

1.2 Ontwikkellingsfases van die projek

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

Fases	Punte	%
Fase 1: Onderzoek en Analise	33	16.5
Fase 2: Ontwerp	30	15
Fase 3: Kodering en Implementering	100	50.0
Fase 4: Dokumentasie en Algemene Evaluering	37	17.5
Totaal	200	100

Dokumentasie/bewys van wat jy gedurende elke fase gedoen het, moet met gespesifiseerde intervalle ingelewer 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 geassesseer gaan word, verband hou met wat in die vorige fases gedoen is.

Sperdatums vir die inlewing 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 volgens die assesseringsinstrumente aangespreek het.

Neem die terugvoering 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** gedoen word en deur die onderwyser gefasiliteer en deurlopend gemonitor word.

Jy moet aandag gee aan die volgende minimum kriteria:

Onderzoek en Analise

- 'n Beskrywing van die probleem in jou eie woorde waarin die hoofaspekte uitgelig word.
- Onderzoek die onderwerp: Die ondersoek 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 potensiele gebruikers in te samel, insluitend hulle behoeftes en enige beperkings wat daar mag wees.
- Analise 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 op 'n geïntegreerde manier insluit. (Ander toepassings kan met hierdie ontwikkelingsinstrumente geïntegreer word).
- Ander aspekte van die programmeringsprojek wat geassesseer sal word:
 - Programmeringstyl
 - Grafiese Gebruikerskoppelvlak (GUI)
 - Gebruik van Mens-Rekenaar-Interaksiebeginsels
 - Vlak van kundige programmering
 - Funksionaliteit van die program
 - Robuustheid van die program, insluitend die gebruik van defensiewe programmeringstegnieke
 - Stem die projek ooreen met die oorspronklike mikpunte en doelwitte?

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

Dokumentering en Algemene Evaluering

- 'n Drukstuk van die bronkode, kommentaar ingesluit, installeringsprosedures asook apparatuur- en programmatuurvereistes – Tegnieke 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 ingelewer?
 - 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 programmeringsprojek

Die instruksies vir die verskillende fases is soos volg:

FASE 1:

Onderzoek en Analise

Sperdatum: _____

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

1. Probleemstelling

- Identifiseer en 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 in 'n werklike konteks of situasie soos dit geïdentifiseer is, ondersoek.
- Jou probleemstelling moet
 - duidelik beskryf wat die probleem behels.
 - die aspekte noem wat opgelos behoort te word.
 - aandui wat die doel van die programmatuur sal wees.

Voorbeeld:

Ek gaan programmatuur ontwikkel wat deur graad 3-leerders gebruik kan word om optelling en aftrekking as deel van hulle gesyferdheidsleerprogram te oefen en in te dril. Op die oomblik word dit per hand gedoen en hierdie stelsel steun op die onderwyser om antwoorde na te gaan en terugvoer aan leerders te gee. Dit is tydrowend en maak dit ook moeilik vir die onderwyser om aandag aan 'n groot hoeveelheid leerders tydens klasure te gee. Die voordeel van die program sal wees dat dit konsepte sal toets, onmiddellike terugvoer aan die leerders sal gee en hulle van die korrekte antwoorde sal voorsien, asook verduidelikings sal gee en hulle met meer gevorderde oefeninge sal laat voortgaan sodra hulle 'n konsep bemeester het. Die onderwysers sal tyd op hande hê om aandag aan leerders te gee wat sukkel en persoonlike aandag en verdere verduideliking nodig het. Die program sal die voorkoms en gevoel van 'n speletjie hê, wat leerders sal geniet. Die program sal in staat wees om leerders se antwoorde na te gaan vir korrektheid, onmiddellike terugvoering te voorsien, rekord te hou en verslag te doen van leerders se punte, sukses en vordering, asook om leerders aan te raai wanneer om na meer gevorderde oefeninge aan te beweeg.

2. Onderzoek

- Onderzoek die onderwerp om feite in te samel en om die aard en die omvang van die probleem vas te stel, bv. wat sal deel van die programmatuur wees en wat sal nie ingesluit wees nie. Die doel is om die vereistes, behoeftes en beperkings vas te stel.
- Jy sal dalk die volgende wil uitvind:
 - Watter onderwysers of skool moontlik die tipe programmatuur wat jy besig is om te ontwikkel, sal gebruik.
 - Watter stelsel of prosesse gebruik word in die afwesigheid van programmatuur (bv. papiergebaseerde stelsel)
 - Inligting wat spesifiek verband hou met jou onderwerp (bv. as jy 'n program ontwikkel vir graad 3-leerders moet jy weet watter reeks getalle hulle gebruik met optelling en aftrekking en of hulle met negatiewe getalle werk, 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 eindgebruikers, maak notas, versamel dokumente en maak opsommings en afleidings uit al hierdie inligting.
- Neem waar hoe die huidige stelsel of soortgelyke stelsel tans werk. Indien moontlik, maak notas.
- Maak notas oor die toevoer, verwerking en afvoer wat vereis sal word.
- Moontlike bewyse wat ingelewer kan word: Vraelyste, notas (of opnames) wat gemaak is tydens onderhoude, foto's, dokumente/vorms wat in die huidige stelsel gebruik word, oudio-/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 terugvoering van die gebruikers en skryf 'n moontlike/beoogde oplossing vir die probleem in jou eie woorde neer.
- Moontlike bewyse wat ingelewer kan word: Definisie van die probleem en/of lys van vereistes. Dit is nie 'n gedetailleerde spesifikasie met toevoer en afvoer nie maar eerder slegs 'n eenvoudige lys van eienskappe wat die beoogde oplossing sal voorsien aan gebruikers.

5. Lewer 'n beplanningsdokument in.

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

- Beskrywing van die probleem
- Agtergrondinligting oor die onderwerp met verwysings
- Sowel bewyse van die metodes wat gebruik is om inligting in te samel as inligting omtrent die gebruikers of die persone wat daaraan deelgeneem het
- Breë oorsig van 'n moontlike oplossing vir 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 aan die vereistes voldoen 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:
 - Datavalidering (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, oorerwingshiërargieë waar van toepassing (hoër orde)
 - Grafiese gebruikerskoppelvlak-ontwerp (GUI)
Tekeninge en/of skermkote (“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. Lewer ’n dokument met al die programspesifikasies soos hierbo aangedui, in.
Die dokument moet die volgende insluit:
 - Toevoer-, verwerkings- en afvoervereistes
 - 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, vloedigramme en diagramme soos EV-diagramme, klasdiagramme en /of skermskote 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:
 - Toepaslike toevoerstrategieë
 - Toegang tot die databasis
 - Toepaslike en effektiewe algoritmes
 - 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 (werk) en geen foute het nie.
 - Gebruik 'n toetsstrategie 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 programmeringsbeginsels 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?

Let op:

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% oorskry nie.**

FASE 4:**Dokumentasie en Evaluering****Sperdatum:** _____

Met voltooiing van hierdie fase moet 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 oorsigtelike oplossing van die probleem soos saamgestel in fase 1
- Databasisontwerp, skermkote van die ontwerpansig (“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/CD-speler
- Spesifikasies van die toetsprosedures en die toetsresultate
- 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
- Apparaat 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 deurstap deur 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. Lewer 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, inlewer.
- 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 vir die projek toe tydens die demonstrasie.
- Die onderwyser moet **willekeurig 'n paar gedeeltes 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/haar eie projek gebruik is nie, kan geen punte vir die projek aan die leerder toegeken word nie.
- Die leerder lewer 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?
- Gebruikswaarde 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?

INLIGTINGSTEGNOLOGIE
PRAKTIESE ASSESSERINGTAAK (PAT)
2009

ASSESSERINGSINSTRUMENTE

ASSESSERINGSINSTRUMENTE VIR DIE PROGRAMMERINGSPROJEK**Assessering vir Fase 1****Naam van leerder:** _____

Onderzoek en Analise: Kriteria					Moontlike punt	Punt behaal	
Probleemstelling	Die probleemstelling					4	
	4	3	2	1	0		
	Die probleem is duidelik gestel en beskryf en ondubbelsinnig – stel duidelik wat die probleem behels. Lê uit watter aspekte opgelos moet word. Duidelike stelling van wat die doel van die programmatuur sal wees.	Die stelling is duidelik maar met geringe tekortkominge.	Die stelling is vaag, laat die leser onseker oor wat die doel van die stelsel is.	Die stelling is so vaag dat geen waarneembare doel gevind kan word nie.	Geen stelling of beskrywing nie.		
Onderzoek	Sleutelareas met betrekking tot die probleem					4	
	4	3	2	1	0		
	Onderzoek definieer/ verduidelik omvattend en duidelik die sleutelareas rakende die onderwerpe Toon goeie insig en begrip van die sleutelareas van die onderwerp	Geringe tekortkominge. Toon insig in meeste van die sleutelareas van die onderwerp	Toon 'n mate van insig in sommige sleutelareas	Vaag. Toon min begrip van sleutelareas	Sleutelareas nie gedefinieer nie		
	Lys van verwysings					2	
	2	1	0				
	Omvattende lys Aanvaarbare formaat	Minder omvattend – geringe tekortkominge	Geen verwysings nie				
Vereistes	Toevoervereistes					3	
	3	2	1	0			
	Omvattende lys van toevoervereistes verkry van gebruikers, goed gedefinieer en verkry deur ten minste van twee verskillende metodes soos vraelyste, onderhoude, gebruik te maak	Minder omvattende lys van toevoervereistes verkry van gebruikers deur gebruik te maak van ten minste een metode. Nie so duidelik gedefinieer nie maar nog steeds aanvaarbaar	Min toevoervereistes verkry van verbruikers of vereistes is opgemaak deur die leerder. Vaag en nie duidelik gedefinieer nie	Geen toevoervereistes verkry nie			

Verwerkingsvereistes						
3	2	1	0			
Omvattende lys van verwerkingvereistes verkry van gebruikers, goed gedefinieerd en verkry deur ten minste van twee verskillende metodes soos vraelyste, onderhoude, gebruik te maak	Minder omvattende lys van verwerkingvereistes verkry van gebruikers deur gebruik te maak van ten minste een metode. Nie so duidelik gedefinieerd nie maar nog steeds aanvaarbaar	Min verwerkingvereistes verkry van gebruikers of vereistes is opgemaak deur die leerder. Vaag en nie duidelik gedefinieerd nie	Geen verwerkingvereistes verkry nie		3	
Afvoervereistes						
3	2	1	0			
Omvattende lys van afvoervereistes verkry van gebruikers, goed gedefinieerd en verkry deur ten minste twee verskillende metodes soos vraelyste, onderhoude, te gebruik	Minder omvattende lys van afvoervereistes verkry van gebruikers deur ten minste een metode te gebruik. Nie so duidelik gedefinieerd nie maar nog steeds aanvaarbaar	Min afvoervereistes verkry van verbruikers of vereistes is opgemaak deur die leerder. Vaag en nie duidelik gedefinieerd nie	Geen afvoervereistes verkry nie		3	
Metodes						
2	1	0				
Ten minste 2 verskillende metodes is gebruik om vereistes te verkry, bv. vraelyste, voorbeeld dokumente, notas/oudio/ video van onderhoude/ ondersoek, ens.	Slegs een metode is gebruik om vereistes te verkry, bv. vraelyste, voorbeeld dokumente, notas/oudio/ video van onderhoude/ ondersoek, ens.	Geen vereistes verkry nie			2	
Bewyse dat inligting ingesamel is rakende vereistes, bv. vraelyste, voorbeelddokumente, notas/oudio/ video van onderhoude/ ondersoek, ens. (1 punt vir elke tipe/metode gebruik vir die insameling van toepaslike bewyse tot 'n maksimum van 3 punte)					3	
Moontlike oplossing						
4	3	2	1	0		
Duidelike en omvattende beskrywing – 'n duidelike oorsig van 'n moontlike oplossing is gegee. Dui duidelik aan wat die program sal insluit en doen.	Duidelik maar met geringe tekortkominge.	'n Basiese beskrywing is gegee maar sommige aspekte van die voorgestelde oplossing is vaag.	Die beskrywing is vaag. Geen duidelike oplossing kan geïdentifiseer word in die beskrywing nie.	Geen moontlike oplossing gegee nie.	4	
Die reikwydte (grense en aannames/eienskappe wat ingesluit sal word)						
3	2	1	0			
Die reikwydte van die voorgestelde oplossing is duidelik en goed gedefinieerd. Toon duidelik aan wat deel sal wees van die program en wat dit nie sal insluit nie.	Gedefinieerd maar sommige geringe tekortkominge. Nie altyd duidelik oor wat die program sal insluit of nie sal insluit nie.	Vaag gedefinieerd. Geen duidelike reikwydte kan bepaal word nie.	Nie gedefinieerd nie.		3	
Toepaslikheid van voorgestelde oplossing in die konteks van die scenario						
2	1	0				
Besonder toepaslik. Goeie toepassing in die scenario	Toepaslik. Toepassing in scenario nie altyd oortuigend nie	Nie toepaslik nie. Toepassing in scenario nie oortuigend nie			2	
Totaal:					33	

Assessering vir Fase 2: Naam van leerder:

Ontwerp: Kriteria				Moontlike punt	Punt behaal	
Spesifikasies	Toevoer				3	
	3	2	1	0		
	Gebruikerstoevoer en ander bronne van toevoer is duidelik beskryf in terme van watter toevoer vereis word, asook die formaat (bv. datumformaat) van toevoer.	Beskryf met geringe tekortkominge in terme van wat vereis word en in watter formaat.	Die beskrywing is vaag of onvolledig.	Slegs gelys – geen beskrywing nie.		
	Verwerking					
	3	2	1	0		
	Beskryf duidelik hoe data verwerk/ gemanipuleer sal word in terme van vereistes, formaat, berekening, formules, ens. Kort, duidelik en korrekte beskrywing van alle gevalle waar toepaslik. Gebruiker het duidelikheid oor resultaat.	Beskrywing van verwerking/ manipulering van data is duidelik in meeste van die gevalle waar dit vereis word. Goeie poging maar kan verbeter. Nie altyd korrek of toepaslik nie.	Beskrywing van verwerking/ manipulering van data nie duidelik in die meeste gevalle nie. Beskrywing is vaag/ onvolledig. Nie korrek/toepaslik in meeste gevalle nie.	Verwerking/ manipulering van data nie beskryf nie.		
	Afvoer				3	
	3	2	1	0		
	Vereiste afvoer (skerm, verslag) is geïdentifiseer en duidelik beskryf volgens die vereistes en in terme van formaat (geldeenheid, meeteenheid, ens.)	Geïdentifiseer en beskryf met geringe tekortkominge.	Die beskrywing is vaag of meeste van die vereiste afvoer is nie geïdentifiseer nie of die formaat van afvoer is nie beskryf nie.	Afvoer nie geïdentifiseer of beskryf nie.		
	Datavalidering				3	
	3	2	1	0		
	Aangedui vir alle toevoer en in detail beskryf. Betekenisvol en effektief.	Geïdentifiseer en beskryf met geringe tekortkominge.	In sommige gevalle aangedui waar van toepassing en/of die beskrywing van validering is vaag of onvolledig.	Nie aangedui nie.		
Foutboodskappe geassosieer met datavalidering				2		
2	1	0				
Aangedui vir alle toepaslike valideringskontrole of foute en in detail beskryf/aangetoon.	Sommige foutboodskappe aangedui en/of beskrywing van foutboodskappe is vaag.	Geen foutboodskappe aangedui vir enige datavaliderings of foute nie.				

Oorsig van databasisontwerp	Velde				2	
	2	1	0			
	Goed gekose velde, veldtipes en -groottes om die toepassing te pas. Veldeienskappe soos validering en maskers is aangetoon waar van toepassing.	Meestal goed gekose velde, veldtipes en -groottes om die toepassing te pas. Veldeienskappe soos validering en maskers nie altyd aangedui waar van toepassing nie.	Velde, veldtipes en -groottes nie goed gekies om toepassing te pas nie. Veldeienskappe sos validering en maskers nie aangedui nie.			
	Tabelle					
3	2	1	0		3	
Velde goed gegroepeer in tabelle. Geen herhaling van velde in tabelle nie. Meer as twee tabelle met korrekte verwantskappe tussen die tabelle. Toon korrekte primêre en vreemde sleutels.	Velde goed gegroepeer in tabelle met geringe tekortkominge. Herhaling van velde in tabelle in sommige gevalle. Ten miste twee tabelle met korrekte verwantskappe tussen tabelle. Toon korrekte primêre en vreemde sleutels.	Velde meestal goed gegroepeer in tabelle/ uitgebreide herhaling van velde in tabelle. Ten minste twee tabelle maar geen of verkeerde verwantskap tussen tabelle. Toon geen of verkeerde primêre en vreemde sleutels.	Slegs een tabel/ slegs velde gegee.			
Datastrukture	Organiseer en stoor data (versamelings met stel data en stel bewerkings wat op data uitgevoer kan word, bv. klasse en objekte)				3	
	3	2	1	0		
	Abstrakte datatipes gebruik. Toepassing goed beplan in terme van klasse en objekte. Objekte goed gestruktureer met toepaslike metodes.	Gebruik abstrakte datatipes. Ruimte vir verbetering. Toepassing nie heeltemal beplan in terme van klasse en objekte. Nie altyd van toepassing nie. Die hele program is nie objekgeoriënteerd nie.	'n Poging om abstrakte datatipes te gebruik. Objekte nie altyd goed saamgestel nie. Objekte saamgestel maar nie korrek gebruik nie/glad nie gebruik nie. Baie min van die program is objekgeoriënteerd.	Geen abstrakte datatipes (klasse en objekte) nie.		
	Toevoer					
3	2	1	0		3	
Toepaslike toevoerkomponente om akkurate/geldige toevoer in alle gevalle waar vereis te verseker. Gemotiveerde keuses waar vereis word.	Toepaslike komponente waar vereis word. Keuses meestal gemotiveer. Geringe tekortkominge.	Toepaslik in meeste gevalle waar vereis word. Keuses nie altyd gemotiveer nie.	Nie gedoen nie/nie toepaslike komponente geïdentifiseer nie.			
GUI ontwerp	Afvoer				3	
	3	2	1	0		
	Toepaslike afvoerkomponente om akkurate/geldige toevoer in alle gevalle waar vereis te verseker. Gemotiveerde keuses waar vereis word.	Toepaslike komponente waar vereis word. Keuses meestal gemotiveer. Geringe tekortkominge.	Toepaslik in meeste gevalle waar vereis word. Keuses nie altyd gemotiveer nie.	Nie gedoen nie/nie toepaslike komponente geïdentifiseer nie.		
	Spreek die vereistes van fase 1 aan					
2	1		0		2	
Alle vereistes aangespreek	Ten minste 50%		Minder as 50%			
Totaal:					30	

Assessering vir Fase 3:

Naam van leerder: _____

NB: Fase 3 en 4 sal slegs geassesseer word wanneer Fases 1 en 2 voltooi is EN Fases 1 en 2 verband hou met die onderwerp gedek in Fases 3 en 4

Kodering en Implementering: Kriteria					Moontlike punt	behaal			
Databasis	Databasistabelle					4			
	4	3	2	1	0				
	Aantal tabelle voorsien 'n effektiewe oplossing – tabelle is genormaliseer (Toepaslike aantal tabelle om 'n doeltreffende oplossing te bied) Primêre en vreemde sleutels in verwante tabelle effektief en toepaslik. Alle verwantskappe is goed gedefinieer deur gebruik van korrekte primêre en vreemde sleutels.	Aantal tabelle bied 'n goeie oplossing. Ten minste een doeltreffende en toepaslike verwantskap wat die korrekte primêre of vreemde sleutels gebruik. Sommige primêre of vreemde sleutels in verwante tabelle is nie toepaslik nie.	Aantal tabelle bied 'n gedeeltelike oplossing – tabelle nie genormaliseer nie (Kon meer tabelle gebruik het vir 'n effektiewer oplossing). Meeste van die primêre of vreemde sleutels nie geskik nie of versteksleutels gebruik.	Aantal tabelle nie relevant vir die oplossing nie. (Slegs een tabel/n aantal irrelevante tabelle met herhaling van velde in die tabelle. Of meer as een databasis met een tabel elk in plaas van een databasis)	Geen databasis gebruik nie.				
	Veldtipes							2	
	2	1		0					
	Alle velde met toepaslike datatipes	Datatipes van sommige velde nie toepaslik nie		Slegs verstektipes gebruik of geen databasis nie					
	Veldgroottes – databasis					2			
	2	1		0					
	Alle velde met toepaslike veldgroottes	Sommige van die velde is te groot/klein		Slegs verstekveldgroottes gebruik of geen databasis nie					
	Beskrywende veldname in databasis					1			
LET OP: Assessee die kode indien die program nie uitvoer nie. Hierdie afdeling het te make met die programmering in Delphi/Java uitsluitend SQL.									
Gebruikersgedefinieerde datastrukture (uitgesluit databasistabelle)					3				
3	2	1	0						
Toepaslike en doeltreffendste datastrukture in alle gevalle gebruik om die probleem op te los.	Toepaslike en doeltreffendste datastrukture in meeste gevalle gebruik om die probleem op te los. Geringe tekortkominge.	Toepaslike en doeltreffendste datastrukture in minder as 50% van gevalle gebruik.	Ontoepaslike of ondoeltreffende gebruik van datastrukture.						
Veranderlikes/datastrukture se name betekenisvol					2				
2	1		0						
Veranderlikes/datastrukture se name is betekenisvol in alle gevalle, regdeur die program.	Mindere tekortkominge.		Nie betekenisvol ni.						
Seleksiestrukture					3				
3	2	1	0						
Toepaslike en doeltreffendste keusestrukture in alle gevalle gebruik om probleem op te los.	Toepaslike en doeltreffendste keusestrukture in meeste gevalle gebruik.	Onvanpaste of ondoeltreffende gebruik van keusestrukture in meeste gevalle.	Nie gebruik nie.						

	Herhalingstrukure				3
	3	2	1	0	
	Toepaslike en doeltreffendste herhalingstrukture in alle gevalle gebruik om probleem op te los.	Toepaslike en doeltreffendste herhalingstrukture in meeste gevalle gebruik.	Onvanpaste of ondoeltreffende gebruik van herhalingstrukture in meeste gevalle.	Nie gebruik nie.	
	Komplekse programmeringstegnieke (vb. Speel videosnitte, geleende kode, rafels (threads), netwerk)				
	3	2	1	0	3
	Dit werk korrek. Toepaslike gebruik en voeg waarde toe tot die oplossing.	Dit werk korrek. Nie altyd toepaslik gebruik nie of voeg nie regtig waarde toe tot die program nie.	Poging aangewend maar dit werk nie behoorlik nie. Ontoepaslik gebruik. Nie relevant vir die oplossing nie.	Geen komplekse tegnieke gebruik nie of oorskry meer as 10% van die kode.	
	Datavloei en prosesse (gebruikersgedefinieerde parameteroordrag)				
	3	2	1	0	
	Uitstekende interaksie/kommunikasie tussen modules/klasse. Sluit gevorderde datatipes soos parameters of terugstuurtipes in.	Vaardige/voldoende/mate van kommunikasie tussen modules/klasse met klein foute. Sluit mate van parameteroordrag tussen modules in.	Beperkte kommunikasie tussen modules/klasse. Slegs primitiewe datatipes as parameters oorgedra.	Geen kommunikasie tussen modules/klasse nie. Geen parameters oorgedra nie.	3
	Hergebruik van kode (klasse en metodes/subprogramme)				
	3	2	1	0	
	Toepaslike en doeltreffende hergebruik van kode en/of metodes Modules onafhanklik. Lokale veranderlikes gebruik waar toepaslik. Globale veranderlikes slegs gebruik wanneer vereis	Hergebruik van kode en of metodes, maar nie altyd toepaslik/effektief nie. Modules nie altyd onafhanklik nie. Sommige globale veranderlikes beperk onafhanklikheid van modules	Hergebruik van kode nie toepaslik/effektief nie Modules kon afgebreek. gewees het in meer modules. Byna alle veranderlikes globaal verklaar	Lineêre programmering – een aaneenlopende program, geen modules nie. Geen hergebruik van kode en/of metodes nie	
	Oplossingalgoritmes				3
	3	2	1	0	
	Alle oplossingalgoritmes gebruik om probleem op te los is toepaslik en doeltreffend	Toepaslik en doeltreffend met geringe tekortkominge	Meestal toepaslik of doeltreffend	Totaal ontoepaslike oplossingalgoritmes of meestal ondoeltreffend	
	Korrektheid van oplossingalgoritmes				
	3	2	1	0	3
	Geen logiese foute nie. Alle verwerkingsresultate is korrek.	Geringe tekortkominge. Baie min logiese foute. Baie min van die verwerkingsresultate is foutief.	Logiese foute, Baie resultate is foutief.	Baie logiese foute. Byna al die resultate is foutief/min van die vereiste resultate word gelewer.	
	Toevoerstrategieë				
	3	2	1	0	
Toevoer	Besonder toepaslik, doeltreffende toevoerstrategieë in alle gevalle, bv. toevoer van teksleërs, databasistabelle, gebruikertoevoer	Toepaslik en doeltreffend met geringe tekortkominge	Sommige strategieë kon geskikter/doeltreffender wees	Meestal ontoepaslik of ondoeltreffend	3
	Afvoer vs. vereistes				
	3	2	1	0	
	Afvoer voldoen aan al die vereistes vir die oplossing	Afvoer voldoen aan meeste van die vereistes vir die oplossing	Afvoer voldoen aan minder as 50% van die vereistes vir die oplossing	Geen afvoer nie	
Afvoer				3	

Struktuur van afvoer (kodering)				
3		2		1
Afvoer altyd goed gestruktureerd, leesbaar met opskrifte en subopskrifte. Opskrifte herhaal op volgende bladsye/skerms waar van toepassing		Afvoer in meeste gevalle goed gestruktureerd, leesbaar met opskrifte en subopskrifte. Opskrifte herhaal in meeste gevalle op volgende bladsye/skerms waar van toepassing		Afvoer nie goed gestruktureerd nie. Opskrifte en/of subopskrifte nie goed geformuleer nie of afwesig. Opskrifte meestal nie op daaropvolgende bladsye waar van toepassing herhaal nie.
Geen kode om afvoer te vertoon nie				3
Formaat van afvoer – alle waardes toepaslik geformateer waar van toepassing, bv. geldeenhede, meeteenhede, ens.				
1				
Databasiskoppeling: string/pad korrek opgestel en dit werk				
1				
Databasisinteraksie met program				
2		1		0
Databasis in wisselwerking met program op 'n betekenisvolle wyse, bv. navrae en verslae		Nie altyd betekenisvol nie		Geen interaksie/nie betekenisvol nie
				2
Rekords korrek gemanipuleer via SQL (2 punte vir elke korrekte SQL-stelling tot 'n maksimum van 6 punte)				
<i>Insert, delete, select, update.</i> Noem ander:				
				6
Manipuleer velde via SQL (2 punte vir elke korrekte SQL-stelling tot 'n maksimum van 6 punte)				
Berekeninge op velde, verander inhoud, vertoon slegs sekere velde, benoemde velde, rangskik volgens veld. Noem ander:				
				6
Program kompyleer suksesvol – geen sintaksfoute nie				
1				
Uitvoerfoute				
3		2		1
Geen uitvoerfoute nie. Alle opsies voer suksesvol uit		Sommige van die opsies lewer foute wanneer uitgevoer word		Slegs een of twee opsies voer suksesvol uit
Geen opsies voer suksesvol uit nie				3
Toevoervalidering				
3		2		1
Alle toevoer wat gevalideer behoort te word, word gevalideer deur gebruik van kode		Meeste van die toevoer wat gevalideer behoort te word, word gevalideer		Minder as 50% van die toevoer wat gevalideer behoort te word, word gevalideer
Geen validering nie				3
Foutboodskappe				
3		2		1
Toepaslik en gebruikersvriendelike foutboodskappe in alle gevalle waar datavalidering toegepas is		Toepaslik en gebruikersvriendelike foutboodskappe in meeste gevalle waar datavalidering toegepas is		Toepaslik en gebruikersvriendelike foutboodskappe in enkele gevalle waar datavalidering toegepas is
Geen foutboodskappe nie				3
Program gee afvoer (afvoer is die resultaat van verwerking, m.a.w. GUI-vensters sonder funksionaliteit klassifiseer nie as afvoer nie)				
2				
Korrektheid van afvoer				
3		2		1
Program gee sowel korrekte as toepaslike afvoer in alle gevalle		Program gee sowel korrekte as toepaslike afvoer in meeste gevalle		Program gee sowel korrekte toepaslike afvoer in minder as 50% van gevalle
Geen afvoer nie of slegs foutiewe afvoer				3
Hantering van uitsonderings (Exception handling)				
2		1		0
Gebruik <i>try ... catch</i> hantering van uitsonderings		Poging aangewend om uitsonderings te hanteer of gebruik <i>if</i> -stellings om fouthantering te doen		Geen poging nie
				2

Grafiese Gebruikerskoppelvlak (GUI)	Verskillende skerms toepaslik gebruik (vensters/paneel/tab sheets, ens.)			1
	Komponente vir toevoer/afvoer gebruik (geïmplementeer soos per Fase 2) OF toepaslike veranderinge gemaak vanaf Fase 2 se terugvoer)			2
	2	1	0	
	Altyd toepaslik, besonder doeltreffend	Geringe tekortkominge	Meestal nie toepaslik nie of ondoeltreffend	
	Byskrifte/lokboodskappe met eksakte formate vir toevoer			2
	2	1	0	
	Konstant toegepas regdeur die hele projek waar vereis word	Toegepas in meeste van gevalle waar vereis word	Meestal nie toegepas nie	
	Konsekvente uitleg (dieselfde voorkoms en gevoel regdeur die program)			2
	2	1	0	
	Vir alle skerms (dieselfde kleure, fonttipes gebruik regdeur program)	Meeste van die skerms (sommige skerms verskillende kleure, fonttipes, ens. – geringe tekort-kominge.	Uitleg meestal nie konsekwent nie	
	Insiggewende afvoerverslae is insiggewend en maklik om te lees en te interpreteer.			2
	2	1	0	
	Alle afvoer, alle skerms is insiggewend en maklik om te lees en te interpreteer (toepaslike fontgrootte, uitleg, kleur, ens.)	Meeste van die afvoer is insiggewend en maklik om te lees en te interpreteer – geringe tekortkominge	Afvoer meestal nie insiggewend of nie maklik om te lees en te interpreteer nie	
	Groepering van toevoer/afvoer			2
	2	1	0	
	Tipe toevoer/afvoer saam gegroep, bv. adresinligting, alle skerms	Tipe toevoer/afvoer meestal saam gegroep – geringe tekortkominge	In meeste gevalle nie saam gegroep nie	
	Navigasie tussen skerms			2
	2	1	0	
Maklik om tussen skerms te navigeer – logiese vloei van gebeure	Maklik en logies met geringe tekortkominge	Nie maklik of logies nie		
Hulp beskikbaar as deel van die program			2	
2	1	0		
Hulp beskikbaar as deel van die program, werk korrek met duidelike instruksies	Nie altyd beskikbaar of duidelik nie of werk nie altyd korrek nie.	Nie beskikbaar nie		
Kontekssensitief			2	
2	1	0		
Konteks-sensitiewe hulp beskikbaar (<i>tool tip</i> - teks ingesluit), toepaslik en effektief	Nie altyd toepaslik of effektief nie	Nie beskikbaar nie		
Ontwerp vs. teikengebruiker			2	
2	1	0		
Ontwerp neem teikengehoor (ouderdom, geletterdheid, beperkte visie, toepaslikheid van beelde, ens.) in ag. Ontwerp geskik vir teikengebruiker	In ag geneem, maar geringe tekortkominge	Heeltemal ontoepaslik		
Interne dokumentasie			2	
2	1	0		
Kode is regdeur program voorsien van kommentaar/verklarende aantekeninge vir maklike interpretasie en verduideliking	Voorsien van kommentaar/verklarende aantekeninge maar nie regdeur program nie	Geen kommentaar/verklarende aantekeninge nie.		
Skeiding van afdelings			2	
2	1	0		
Afdelings in kode van program duidelik geskei om leesbaarheid te verhoog (spasiëring, kommentaar op metode/onderafdeling, ens.)	Sommige afdelings geskei	Nie geskei nie		
Totaal:			100	

Assessering vir Fase 4:

Naam van leerder:

Dokumentasie en Evaluering: Kriteria				Moontlike punt	Punt behaal	
Tegniese dokumentasie	Databasisontwerp word duidelik aangetoon en verduidelik			1		
	Beskrywing van ander datastrukture wat gebruik is			1		
	Beskryf/Toon verwantskap tussen modules/programme			1		
	Beskrywing van ongewone kode – funksies, berekeninge, geleende kode, ens.			1		
	Spesifikasie van toetsprosedures en toetsresultate			1		
	Fouterstel en foutopsproing word duidelik beskryf			1		
	Voorbeelde van toetslopias met resultate			1		
	Bronkode is beskikbaar			1		
	Lys van datalêers waaruit die program bestaan, word voorsien			1		
	Gebruikersgids	Titelblad			1	
Inhoudsopgawe			1			
Agtergrond van die projek (persoonlike motivering en keuse)			1			
Inleiding tot die projek (genre)			1			
Projek se reikwydte en beperkinge duidelik beskryf			1			
Vereistes/formate vir gebruikerstoevoer			1			
Afvoer/verslae voorsien (skermskote ingesluit)			1			
Gedetailleerde instruksies en deurstep-gids						
		3	2	1	0	
Dui duidelik alle stappe wat gevolg moet word wanneer program gebruik word, aan en dit word ondersteun deur skermskote waar nodig. Goed gestruktureerd met logiese vloei. Goed geformuleer – maklik om te verstaan		Dui meeste van die stappe wat gevolg moet word wanneer die program gebruik word, aan en voorsien soms skermskote waar nodig. Nie altyd goed gestruktureerd en nie altyd 'n logiese vloei nie. Nie altyd goed geformuleer nie	Dui slegs 'n paar van die stappe wat gevolg moet word wanneer program gebruik word, aan. Geen skermskote nie. Nie goed gestruktureerd nie. Nie goed geformuleer nie	Nie gedoen nie	3	
Apparatuur benodig om program uit te voer (1 punt elk tot 'n maksimum van 3 punte)						
Hardeskyfspasie, geheue, SVE, gespesialiseerde toestelle – Gee 'n lys. Lys:			3			
Programmatuurvereistes opgenoem			1			
Installeringsinstruksies voorsien			1			
Lys van vereiste lêers			1			
Gebruikers-foutopspeuring duidelik beskryf			1			
Toekomstige ontwikkelings/uitbreidings beskryf			1			
Erkenning waar benodig			1			
Algemene evaluering	Tydsbestuur					
		3	2	1	0	
	Alle spertye gehaal met al 4 fases en al die vereiste werk was gedoen	Haal 3 spertye of dien op tyd in maar sommige werk was nie gedoen nie	Haal slegs 2 spertye of dien betyds in maar meeste werk nie gedoen nie	Altyd laat, nooit gedoen nie	3	
	Vermoë om kode wat lukraak gekies is se doel en werking te verduidelik					
		3	2	1	0	
Verduidelik alle gekose kode duidelik en met selfvertroue. Toon uitstekende insig	Verduidelik gekose kode met geringe tekortkominge. Toon insig	Nie in staat om meeste van die gekose kode toepaslik te verduidelik nie. Gebrek aan insig	Nie in staat om te verduidelik nie	3		

Toepassing van stelsel in werklike situasie				3	
3	2	1	0		
Die oplossing is 'n werkende stelsel wat in 'n werklike situasie gebruik kan word	Die oplossing is 'n stelsel wat in 'n werklike situasie gebruik kan word met bietjie aanpassing	Sommige dele kan in 'n werklike situasie gebruik word	Totaal irrelevant Sal nie in 'n werklike situasie werk nie		
Totaal:				37	