



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

INLIGTINGSTEGNOLOGIE

RIGLYNE VIR PRAKTIESE ASSESSERINGSTAKE

2013

Hierdie riglyne bestaan uit 28 bladsye.

Inligtingstechnologie

Praktiese Assesseringstaak (PAT)

2013

Riglyne vir die onderwyser

Hierdie afdeling bestaan uit 7 bladsye.

A Inleiding

Die doel van die Praktiese Assesseringstaak (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 en vaardighede.

In Inligtingstegnologie (IT) tel die PAT 25% van die totale promosiepunte vir die vak (d.i. 100 uit 400 punte). Dit word oor die eerste drie kwartale van die graad 12-skooljaar geïmplementeer en moet as een uitgebreide taak onderneem word wat in verskillende fases of 'n reeks kleiner aktiwiteite opgedeel word.

Die IT-PAT bestaan uit drie fases:

- Ondersoek en analise
- Ontwerp
- Kodering, implementering en interne dokumentasie, insluitend aanbieding en algemene evaluering

B Programmeringsprojek

1. Waaroor gaan die programmeringsprojek?

Die programmeringsprojek van die PAT in Inligtingstegnologie vereis dat die leerders die ontwikkelingsinstrumente wat hulle bestudeer het, moet gebruik om 'n programmatuur-oplossing vir 'n spesifieke probleem binne 'n gegewe scenario te ontwikkel.

2. Beplanning vir die programmeringsprojek

2.1 Beskrywing

Scenario vir die 2013-graad 12-PAT:

Daar word van die leerder verwag om 'n program te skryf wat gebruik kan word om een van die volgende te bestuur:

- 'n Verkiesingsproses
- 'n Kompetisie om 'n wenner te identifiseer

Voorbeelde:

- VRL-verkiesings
- Kompetisies, bv. Idols, Mej. Hoërskool, Snaaksste Video, ens.
- Skoolbeheerliggaamverkiesings
- Plaaslike regeringsverkiesings
- Enige ander stemproses/verkiesingsproses wat bestuur moet word om 'n wenner te identifiseer of 'n groep verteenwoordigers te kies.

Die fokus van die stelsel moet wees om die prosesse (soos registrasie, stemming en uitslae) te bestuur en om toepaslike statistieke te voorsien.

Die leerder moet navorsing doen oor stem-/verkiesingsprosesse of kompetisieprosesse, die inligtingsbehoefte, verwerking wat gedoen moet word, statistieke/verslae wat gegenereer moet word, ens. 'n Funksionele en bruikbare stelsel moet ontwerp word wat suksesvol geïmplementeer kan word om stem-/verkiesingsprosesse of kompetisieprosesse te bestuur.

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

Met die voltooiing van die projek sal die leerder die volgende vaardighede 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 assesseringsinstrument sal projekte penaliseer wat nie met die gegewe scenario verband hou nie.

2.2 Ontwikkelingsfases van die projek

Die programmeringsprojek moet in drie fases voltooi word, soos in die tabel hieronder aangedui.

FASE	PUNTE	%
Fase 1: Ondersoek en analise	30	17,5%
Fase 2: Ontwerp	30	17,5%
Fase 3: Kodering, implementering en interne dokumentasie, insluitend aanbieding en algemene evaluering	110	65%
TOTAAL	170	100

Dokumentasie/Bewyse van wat die leerder gedurende elke fase van die ontwikkeling gedoen het, moet op spesifieke tye ingelewer word. Dit is noodsaaklik dat bewys gelewer word van al die fases van die werk. Die dokumentasie wat vir elk van hierdie fases vereis word, word in die leerderafdeling van hierdie dokument gespesifiseer.

Sperdatums vir die inlewing van die finale produk van elke fase sal deur die onderwyser vasgestel word met inagneming van die modereringsdatums vir die verskillende fases. Die produkte van elke fase moet geassesseer word en die punte moet aangeteken word.

2.3 Vereistes van die projek

Die leerders moet aan die volgende minimum kriteria voldoen:

Fase 1: Ondersoek en analise

- Beskrywing van die probleem in die leerder se eie woorde, waarin die hoofaspekte in een paragraaf omskryf/uitgelig word
- Ondersoek van die onderwerp deur:
 - Die aard en die omvang van die probleem te identifiseer en feite te versamel
 - Inligting te kry uit 'n werklike situasie, waarin die programmatuurproduk deur die eindgebruiker gebruik gaan word

- Analisering van die probleem: Wat is die vereistes en wat moet deur die programmeringsoplossing voorsien word?

(Sien Leerderafdeling 2 Fase 1 en Assesseringsinstrument Fase 1)

Fase 2: Ontwerp

- Ontwerp 'n oplossing: Hoe sal die program/stelsel aan die vereistes voldoen? Voorsien 'n goed beplande oplossing vir die probleem.

(Sien Leerderafdeling 2 Fase 2 en Assesseringsinstrument Fase 2)

Fase 3: Kodering, implementering en interne dokumentasie

- Die projek moet die belangrikste ontwikkelingsinstrumente insluit, m.a.w. 'n databasis en programmeringstaal wat toepaslik geïntegreer is. (Ander toepassings kan saam met hierdie ontwikkelingsinstrumente geïntegreer word.)
- Ander aspekte van die programmeringsprojek wat geassesseer sal word, sluit in:
 - Programmeringstyl
 - Grafiese gebruikerskoppelvlak ('GUI')
 - Gebruik van mens-rekenaar-interaksiebeginsels ('HCI') en programmatuur-ingenieurswesebeginsels
 - Funksionaliteit van die program
 - Vlak van programmeringskundigheid
 - Robuustheid van die program, insluitend die gebruik van defensiewe programmeringstegnieke
 - Of die projek ooreenstem met die oorspronklike mikpunte en doelwitte
- Interne dokumentering om dele van die program te verduidelik

Aanbieding en algemene evaluering

- Evalueer die volgende:
 - Tydbestuur van die leerder – Het hy/sy al die sperdatums nagekom?
 - Die leerder se vermoë om dele van kode wat ewekansig gekies word, te verduidelik
 - Geskiktheid van die oplossing binne die konteks van die scenario

(Sien Leerderafdeling 2 Fase 3 en Assesseringsinstrument Fase 3)

3. Instruksies vir die leerders

Verwys na die Leerderafdeling van hierdie dokument.

4. Hulpbronne

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

- Toegang tot 'n rekenaar met die volgende programme:
 - Programmeringstaal: Java of Delphi
 - Kantoortoepassings, soos 'n woordverwerker (bv. MS Word) en 'n aanbiedingspakket (bv. MS PowerPoint)
 - Databasisprogrammatuur, soos MS Access of MySQL
- IDE (vir Delphi is dit deel van die programmeringstaal, maar vir Java sal jy addisionele programmatuur nodig hê, soos Netbeans)

5. Assessering van die PAT

Die projek moet onder gekontroleerde toestande voltooi word. Dit beteken dat dit deur die onderwyser gefasiliteer en deurlopend gemonitor moet word om te verseker dat die werk dié van die leerder is.

Sien Asseseringsinstrumentafdeling vir die assesseringstate vir die verskillende fases.

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

- Die 'Instruksies vir die leerders'-afdeling wat in hierdie dokument ingesluit is
- Die assesseringstate 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 om aandag te gee aan die foute wat hulle gemaak het, voordat hulle met die volgende fase voortgaan. Die punte wat aanvanklik vir die evaluering van 'n spesifieke fase toegeken is, sal NIE na die nabetragting hersien word NIE.

Daar sal van leerders verwag word om hulle stelsel aan die einde van Fase 3 te demonstreer. Onderwysers moet die projekte volgens die assesseringsinstrument wat vir Fase 3 voorsien is, 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 Fase 2 voltooi is, moet die leerder die vroeëre fases vir die nuwe onderwerp oordoen. In hierdie geval sal die onderwyser NIE weer die opgedateerde Fase 1 en/of Fase 2 assesseer NIE. Die punte vir die oorspronklike fases moet aangeteken word. Fase 3 sal egter NIE geassesseer word tensy Fase 1 en Fase 2 navorsing weerspieël wat vir die nuwe onderwerp gedoen is nie.

Die verband tussen al die fases van 'n projek moet streng en deurlopend tydens assessering gekontroleer word. Bewyse van werk wat in vorige fases gedoen is, moet altyd gedurende elke assessering en moderering van 'n spesifieke fase van ontwikkeling beskikbaar wees.

Riglyne vir die demonstrasie en interne evaluering van die projek:

- Die onderwyser moet datums en tye vir demonstrasies skeduleer. Laat ongeveer 15 minute per projek toe vir die demonstrasie, asook ongeveer 5 minute vir die opstelling van die projek en terugvoer van die onderwyser na die tyd.
- Die ontwikkeling van die projek is 'n deurlopende proses. Die onderwyser moet altyd kyk na die werk wat in die vorige fases gedoen is, wanneer 'n spesifieke fase van ontwikkeling geassesseer word. Die onderwyser moet die vordering van die projek, in verhouding met die werk wat in die vorige fases gedoen is, deeglik monitor.
 - Die vereistes wat in Fase 1 geïdentifiseer en uitgespel is, moet in die ontwerpfasie van die projek, d.w.s. Fase 2, gereflekteer word.
 - Die werk wat in die ontwerpfasie, d.w.s. Fase 2, gedoen is, moet in die koderings- en implementeringsfasie van die projek, d.w.s. Fase 3, gereflekteer word.
- Die leerder moet al die vorige dokumentasie (d.i. Fase 1- en Fase 2-dokumentasie) beskikbaar hê wanneer die demonstrasie as deel van Fase 3 plaasvind.
- Die demonstrasies moet elektronies op die rekenaar gedoen word.
- Die leerder moet sy/haar rekenaarprogram uitvoer en al die eienskappe van die program aan die onderwyser toon vir assessering.
- **Leerders mag programmeringskode gebruik wat nie in die standaardweergawe van Java of Delphi voorsien word nie. Dit mag kode wees wat hulle van 'n ander bron, soos 'n boek, die Internet of 'n ander program, gekry het. Hierdie kode mag nie 10% van al die kode oorskry nie.**

- Die onderwyser moet **ewekansig 'n paar gedeeltes van die programmeringskode (uitsluitend die 10% geleende kode) in die projek identifiseer. Die leerder moet dan die doel en die werking van die ewekansig-geselekteerde kode aan die onderwyser verduidelik.** Dit word gedoen om seker te maak dat leerders self die kode skryf. 'n Soortgelyke prosedure sal ook tydens eksterne moderering gevolg word. Indien die leerder nie die kode wat in die projek gebruik is, kan verduidelik nie, kan **geen punte vir die projek aan die leerder toegeken word nie.**
- Die onderwyser moet die leerder vra om die toetsstrategieë uit te voer om al die fasette van die program te toets.
- Die onderwyser kan van die leerder verwag om ander, addisionele toetsprosedures uit te voer om seker te maak dat die hele program korrek werk.
- Die onderwyser moet die nasienblad vir Fase 3 gebruik om tydens die demonstrasie punte toe te ken.
- Die leerder moet die elektroniese weergawe van die projek wat gedemonstreer is, inlewer. Die onderwyser sal hierdie weergawe gebruik om enige uitstaande punte toe te ken sodat die punt gefinaliseer kan word.

6. Rekordhouding en rapportering

Vir elke fase sal die onderwyser die fase assesser, 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.

LET WEL: Sodra die produk van 'n fase ingehandig en geassesseer is, sal daardie fase nie weer geassesseer word nie.

7. Riglyne vir die bestuur van die PAT

Daar is twee maniere waarop die PAT bestuur kan word:

Opsie 1:

- Die onderwyser kan een of twee periodes per week aan die PAT afstaan terwyl hy/sy voortgaan met normale onderrig gedurende die res van die week om die graad 12-kurrikulum af te handel.
- As die onderwyser hierdie opsie kies, moet hy/sy so gou as moontlik gedurende die eerste kwartaal met die PAT begin en een fase per kwartaal voltooi.

Opsie 2:

- Die onderwyser kan 'n aaneenlopende tydperk aan die PAT afstaan, bv. die laaste gedeelte van elke kwartaal en een fase per kwartaal voltooi of eers die graad 12-kurrikulum afhandel en dan 'n aaneenlopende tydperk in die derde kwartaal gebruik.

Die onderwyser moet sy/haar werkskedule volgens die opsie wat hy/sy verkies, beplan.

Daar word aanbeveel dat die onderwyser die leerders se onderwerpe aanteken wanneer hulle met Fase 1 begin om te voorkom dat 'kitsprojekte', wat moontlik nie die leerder se eie werk is nie, kan opduik. Onderwysers moet ook seker maak dat leerders projekte 'registreer' wat hulle in staat is om te voltooi, om sodoende te voorkom dat onderwerpe verander word omdat die aanvanklike onderwerp te moeilik of tydrowend is.

8. WENKE

- Verduidelik eers die PAT en gee 'n oorsig oor die proses van ontwikkeling aan die leerders voordat hulle met Fase 1 begin.
- Bespreek die taak en die onderwerp met die leerders. Laat hulle toe om vrae te vra en maak seker dat hulle die probleem wat opgelos moet word, duidelik verstaan.
- Bespreek voorbeelde van moontlike toepassings binne die gegewe scenario met die leerders. Laat die leerders self met 'n paar idees vorendag kom en bespreek die geskiktheid daarvan.
- Alhoewel 'n ander scenario gebruik is, kan dit 'n goeie idee wees om goeie en swak voorbeelde van PAT'e van vorige jare vir leerders te wys.
- Wees baie streng met die inlewering van die dokumentasie wat in elke fase vereis word.

Inligtingstegnologie
Praktiese Assesseringstaak (PAT)
2013

Riglyne vir die leerder

Hierdie afdeling bestaan uit 10 bladsye.

Inligtingstegnologie – Praktiese Assesseringstaak (PAT)

Die Inligtingstegnologie (IT) -PAT bestaan uit drie fases:

- Ondersoek en analise
- Ontwerp
- Kodering, implementering en interne dokumentasie, insluitend aanbieding en algemene evaluering

1. Beplanning vir die programmeringsprojek

1.1 Beskrywing

Scenario vir die 2013-grad 12-PAT:

Daar word van jou verwag om 'n program te skryf wat gebruik kan word om een van die volgende te bestuur:

- 'n Verkiesingsproses
- 'n Kompetisie om 'n wenner te identifiseer

Voorbeelde:

- VRL-verkiesings
- Kompetisies, bv. Idols, Mej. Hoërskool, Snaaksste Video, ens.
- Skoolbeheerliggaamverkiesings
- Plaaslike regeringsverkiesings
- Enige ander stemproses/verkiesingsproses wat bestuur moet word om 'n wenner te identifiseer of 'n groep verteenwoordigers te kies.

Die fokus van die stelsel moet wees om die prosesse (soos registrasie van kandidate en kiesers, stemming en uitslae) te bestuur en om toepaslike statistieke te voorsien.

Jy moet navorsing doen oor stem-/verkiesingsprosesse of kompetisieprosesse, die inligtingsbehoefte, verwerking wat gedoen moet word, statistieke/verslae wat gegenereer moet word, ens. 'n Funksionele en bruikbare stelsel moet ontwerp word wat suksesvol geïmplementeer kan word om stem-/verkiesingsprosesse of kompetisieprosesse te bestuur.

Jy mag jou eie onderwerp/toepassing binne die konteks van die gegewe scenario kies.

Met die voltooiing van die projek sal jy die volgende vaardighede 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 assesseringsinstrument sal projekte penaliseer wat nie met die gegewe scenario verband hou nie.

LET WEL: Jou finale program moet uit EEN enkele, logies-verwante deel programmatuur bestaan. Projekte wat uit twee of meer nieverwante programme bestaan, sal slegs vir een van die gedeeltes punte kry omdat slegs een van die programme as die werklike projek beskou word.

1.2 Ontwikkelingsfasies van die projek

Die programmeringsprojek moet in drie fasies voltooi word, soos in die tabel hieronder aangedui.

FASES	PUNTE	%
Fase 1: Ondersoek en analise	30	17,5%
Fase 2: Ontwerp	30	17,5%
Fase 3: Kodering, implementering en interne dokumentasie, insluitend aanbieding en evaluering	110	65%
TOTAAL	170	100

Dokumentasie/Bewyse van wat jy gedurende elke fase van die ontwikkeling gedoen het, moet op spesifieke tye ingelewer word. Die bewyse en afvoer van elke fase word hieronder bespreek. Punte kan slegs toegeken word as jy hierdie bewyse aan jou onderwyser voorsien en as die werk wat geassesseer word, verband hou met die werk wat in vorige fasies gedoen is.

Datums vir die inlewering van die dokumentasie/bewyse van die werk wat gedurende elke fase gedoen is, sal deur die onderwyser vasgestel word.

Bestudeer die assesseringsinstrumente vooraf om seker te maak dat jy aandag gee aan al die relevante vereistes volgens die assesseringsinstrumente.

Neem die terugvoering van die onderwyser, aangedui 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 fasies nie reg gedoen is nie.

Al die dokumentasie van die vorige fasies moet tydens elke assessering vir die onderwyser beskikbaar wees.

1.3 Hulpbronne wat vir die projek benodig word

Jy sal ten minste die volgende hulpbronne nodig hê om die projek te voltooi:

- Toegang tot 'n rekenaar met die volgende programme:
 - Programmeringstaal: Java of Delphi
 - Kantoortoepassings, soos 'n woordverwerker (bv. MS Word) en 'n aanbiedingspakket (bv. MS PowerPoint)
 - Databasisprogrammatuur, soos MS Access of MySQL
- IDE (vir Delphi is dit deel van die programmeringstaal, maar vir Java het jy addisionele programmatuur nodig, soos Netbeans)

Die projek moet onder **gekontroleerde toestande** voltooi word en deur die onderwyser gefasiliteer en deurlopend gemonitor word.

Jy moet aan die volgende minimum kriteria voldoen:

Fase 1: Onderzoek en analise

- Beskrywing van die probleem in jou eie woorde, waarin die hoofaspekte in een paragraaf omskryf/uitgelig word
- Onderzoek van die onderwerp deur:
 - Die aard en die omvang van die probleem te identifiseer en feite van potensiële gebruikers, insluitend hulle behoeftes en enige beperkinge wat hulle mag hê, te versamel
 - Inligting te kry uit 'n werklike situasie, waarin die programmatuurprodukt deur die eindgebruiker gebruik gaan word
- Analisering van die probleem: Wat is die vereistes en wat moet deur die programmeringsoplossing voorsien word?

(Sien die volgende afdeling Fase 1 en Assesseringsinstrument Fase 1)

Fase 2: Ontwerp

- Ontwerp 'n oplossing: Hoe sal die program/stelsel aan die vereistes voldoen? Voorsien 'n goed beplande oplossing vir die probleem in terme van:
 - Toevoer, verwerking en afvoer
 - Struktuur en inhoud van die databasis
 - 'GUI' en die vloei van gebeure

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

Fase 3: Kodering, implementering en interne dokumentering

- Die projek moet die belangrikste ontwikkelingsinstrumente insluit, m.a.w. 'n databasisontwerp en programmering wat toepaslik geïntegreer is. (Ander toepassings kan saam met hierdie ontwikkelingsinstrumente geïntegreer word.)
- Ander aspekte van die programmeringsprojek wat geassesseer sal word, sluit in:
 - Programmeringstyl
 - Grafiese gebruikerskoppelvlak ('GUI')
 - Gebruik van mens-rekenaar-interaksiebeginsels ('HCI') en programmatuur-ingenieurswesebeginsels
 - Funksionaliteit van die program
 - Vlak van programmeringskundigheid
 - Robuustheid van die program, insluitend die gebruik van defensiewe programmeringstegnieke
 - Of die projek ooreenstem met die oorspronklike mikpunte en doelwitte
- Interne dokumentering om dele van die program te verduidelik

Aanbieding en algemene evaluering

- 'n Drukstuk van die bronkode, kommentaar ingesluit (interne dokumentering)
- Demonstrasie van die finale produk en ondervraging (deur die onderwyser)
- Die onderwyser sal die volgende evalueer:
 - Jou tydbestuur – Het jy al die sperdatums nagekom?
 - Jou vermoë om dele van kode wat ewekansig gekies word, te verduidelik
 - Die geskiktheid van die oplossing binne die konteks van die scenario

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

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 'n bietjie agtergrondinligting oor 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 taak en die probleem is wat opgelos moet word. Hierdie beskrywing moet nie 'n beskrywing van enige rekenaarkode of die oplossing wees nie. Al wat vereis word, is 'n beskrywing van die probleem wat jy in die werklike konteks of situasie waarin dit geïdentifiseer is, ondersoek.
- Jou probleemstelling moet:
 - Duidelik sê wat die probleem behels (verskaf kortliks agtergrondinligting, m.a.w. hoekom jy die onderwerp gekies het)
 - Sê wat jy probeer doen/oplos en vir wie
 - Die aspekte omskryf wat opgelos moet word/verwagtinge van die programmatuur omskryf
 - Aandui wat die doel van die programmatuur sal wees

2. Onderzoek

- Onderzoek die onderwerp om 'n paar feite te versamel 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 word nie. Die doel is om vereistes, behoeftes en beperkings vas te stel.
- Jy sal dalk die volgende wil uitvind:
 - Watter stelsels of prosesse in die afwesigheid van programmatuur gebruik word (bv. papiergebaseerde oplossings) en wat die beperkinge/uitdagings hiermee is
 - Die sleutelareas waaraan aandag gegee/wat opgelos gaan word
 - Inligting wat spesifiek met jou onderwerp verband hou en waarvoor jy 'n oplossing moet konstrueer (bv. hoe 'n verkiesingsproses werk, hoe om die wenner in 'n kompetisie d.m.v. publieke stemme te bepaal, ens.) en hoe of waar/van wie jy hierdie inligting kan bekom
- Demografiese inligting oor gebruikers van jou stelsel (ouderdom, geslag, gestremdhede, vlak van geletterdheid, ens.)
- Jy moet 'n volledige lys van verwysings byhou van al die hulpbronne (webblaaie, boeke) wat gebruik is.
- Bewys van navorsing: Goed geformateerde en goed voorgestelde opsomming van al die bevindinge van die navorsing.

3. Bepaal vereistes vir die program

- Hou besprekings met eindgebruikers, maak notas, versamel dokumente en maak opsommings en afleidings uit hierdie inligting.
- Neem waar hoe die stelsel of 'n soortgelyke stelsel tans werk. Indien moontlik, maak notas.
- Maak notas oor toevoer, verwerking en afvoer wat vereis sal word.

- Moontlike bewyse wat ingelewer kan word: Vraelyste, notas (of opnames) wat gedurende onderhoude gemaak is, foto's, dokumente/vorms van die huidige stelsel, audio-/video-opnames van onderhoude/hoe die stelsel tans werk, ens.
- Inligting moet verskaf word oor die besonderhede van die onderhoude, vraelyste, gebruikers, ens. Inligting soos die tyd en datum wanneer die vraelys/onderhoud gedoen is, asook die name, ligging en kontakbesonderhede van deelnemers. Jou onderwyser kan hierdie inligting gebruik om te verifieer dat die ondersoek gedoen is op die manier waarop dit aangedui is.

4. Moontlike oplossing

- Gebruik jou navorsing en die terugvoer van die gebruikers en skryf 'n moontlike/beoogde bondige oplossing vir die probleem in jou eie woorde.
- Moontlike bewyse om ingelewer te word: Definisie van die probleem en/of lys van vereistes. Dit is nie 'n gedetailleerde spesifikasie met toevoer en afvoer nie, maar eerder 'n eenvoudige lys eienskappe wat die beoogde oplossing aan gebruikers sal voorsien.

5. Lewer 'n beplanningsdokument in

Kombineer die resultate van jou ondersoek in 'n beplanningsdokument wat bestaan uit:

- Beskrywing van die probleem
- Agtergrondinligting oor die onderwerp met verwysings
- Bewyse van inligting wat versamel is
- 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 daargestel is. **Bestudeer die assesseringsinstrument vir Fase 2 en 3 om seker te maak dat jou projek voldoen aan die vereistes waaraan alle take moet voldoen.**

1. Skakel vereistes/eienskappe om in spesifikasies:
Spesifiseer die oplossing in terme van:
 - Toevoer
 - Verwerking
 - Afvoer (tabelle, grafieke, persentasies, ens. ingesluit)Sluit die volgende in as deel van die spesifikasies:
 - Defensiewe programmeringstegnieke soos:
 - Datavalidering (dui aan waar en hoe dit toegepas moet word)
 - Foutboodskappe
 - Datastrukture om data te organiseer en te stoor:
 - Databasisontwerp
 - Velde (tipes, name en groottes)
 - Verwantskappe
 - Sleutels (primêr en vreemd)
 - ER-diagramme
 - Datatipes en strukture wat vir programmering gebruik gaan word:
 - Primitiewe datatipes – enkele waarde, bv. 'integer', 'real/double'
 - Gestruktureerde datatipes – versamelings van data, bv. skikkings, matrikse, rekords, stelle ('sets'), kombinasielyste ('combo boxes'), lyste, vektore
 - Abstrakte datatipes – versamelings van stel data en stel bewerkings wat op data uitgevoer kan word, bv. klasse en objekte
 - Ontwerp van klasse – metodes met parameters en terugstuur-datatipes, attribute, konstruktors, klasdiagramme, oorerwingshiërgieë waar van toepassing (hoër orde)
 - Grafiese gebruikerskoppelvlak ('GUI') -ontwerp
Tekeninge en/of skermkopieë ('screen shots') van die 'GUI' wat die volgende aandui:
 - Uitleg van die komponente
 - Navigering (bv. Hoe beweeg die gebruiker van een skerm na die volgende 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 tussen modules gestuur?)
 - Motivering vir die gebruik van toevoer- en afvoerkomponente (bv. Hoekom is 'n kombinasielys ('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 in met al die programspesifikasies soos hierbo genoem. Die dokument moet die volgende insluit:
- Toevoer-, verwerkings- en afvoervereistes
 - Fout-toetsing en valideringsprosedures
 - Databasisontwerp
 - Lys van datastrukture wat gebruik is (insluitend klasdiagramme)
 - Ontwerp en uitleg van die 'GUI'

Die dokument kan TVA-tabelle, vloei-diagramme en diagramme soos EV-diagramme, klasdiagramme insluit en gevaldiagramme ('case diagrams') en/of skermafdrukke of enige ander geskikte instrumente gebruik wat 'n duidelike voorstelling van die stelsel sal gee, insluitend die nodige aantekeninge en beskrywings van spesifikasies.

FASE 3:

Kodering en Implementering

Sperdatum: _____

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

Voorgestelde stappe om hierdie fase te voltooi:

1. Breek die oplossing, soos omskryf in Fase 2, in modules (bv. volgens opsies) op.
2. Kodeer/Skep die 'GUI'.
3. Skep die datastrukture om die data te organiseer en te stoor.
4. Kodering – Kodeer die oplossing volgens vereistes en spesifikasies (toevoer, verwerking en afvoer) wat in die vorige fase saamgestel ('compiled') is. Punte sal onder andere toegeken word vir die volgende:
 - 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 en geen foute het nie.
 - Gebruik 'n toetsstrategie om te verseker dat die program/stelsel doen wat dit behoort te doen deur verskillende reekse data te gebruik, insluitend ekstreme/foutiewe toetsdata. Hou boek van hierdie data aangesien dit in jou dokumentering benodig sal word.
7. Interne dokumentering
8. Evalueer jou program/stelsel – Doen die program wat dit moet doen/Voldoen die program 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?
 - Voldoen die program aan dit wat in Fase 1 en 2 aangedui is?
 - Is die 'GUI' goed ontwerp?

LET WEL: Jy word toegelaat om gebruik te maak van geleende kode, maar dit mag nie meer as 10% van jou programmeringskode oorskry nie. Geleende kode is enige programmeringskode wat nie deur die standaardweergawe van Java of Delphi voorsien word nie en wat nie deur jou self geskryf is nie. Dit sal tipies kode wees om ongewone funksies uit te voer, soos om 'n video-uittreksel te speel, ens. **Jou program sal nie deur die onderwyser aanvaar word indien dit die geleende kode se perk van 10% oorskry nie.**

9. Aanbieding en evaluering

9.1 **Lewer in:**

- Elektroniese kopie van programme en al die betrokke lêers, soos data lêers, ens.

9.2 **Demonstreer die program vir evaluering en ondervraging**

Riglyne vir die demonstrasie van die projek:

- Die onderwyser sal datums en tye vir demonstrasies skeduleer. Ongeveer 20 minute sal per projek toegelaat word.
- Jy moet al die dokumentasie ten minste EEN week voordat die demonstrasie plaasvind, inlewer.
- Die demonstrasie moet elektronies op die rekenaar gedoen word.
- Jy moet jou rekenaarprogram uitvoer en al die eienskappe van die program aan die onderwyser toon vir evaluering.
- Die onderwyser kan 'n toetsstrategie wat gegee word, as 'n riglyn gebruik en jou vra om dele van of al die toetsprosedures uit te voer.
- Die onderwyser kan jou vra om addisionele toetsprosedures uit te voer om seker te maak dat die hele program korrek werk.
- Die onderwyser sal die nasienblad vir Fase 3 as riglyn gebruik en punte daarvolgens toeken tydens die demonstrasie.
- Die onderwyser sal willekeurig 'n paar gedeeltes programmeringskode (uitsluitend die 10% geleende kode) in die projek identifiseer as deel van die demonstrasie. Jy moet dan die doel en werking van die willekeurig gekose kode aan die onderwyser verduidelik. Dit word gedoen om seker te maak dat jy alle kode self geskryf het. 'n Soortgelyke prosedure sal tydens eksterne moderering gevolg word. Indien jy nie die kode wat in jou projek gebruik is, kan verduidelik nie, kan geen punte vir die projek toegeken word nie.
- Jy moet na afloop van die demonstrasie die elektroniese weergawe van die projek inlewer. Die onderwyser sal hierdie weergawe gebruik om enige uitstaande punte toe te ken sodat die punt gefinaliseer kan word.

9.3 **Finale algemene evaluering**

Die onderwyser sal die volgende evalueer:

- Tydbestuur – Het jy al die sperdatums nagekom?
- Gebruikswaarde – 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 'n aaneenlopende proses?

Inligtingstechnologie
Praktiese Assesseringstaak (PAT)
2013

Assesseringsinstrumente

Hierdie afdeling bestaan uit 10 bladsye.

Assesseringsinstrumente vir die programmeringsprojek

Assessering vir Fase 1

Naam van leerder: _____

Onderzoek en Analise: Kriteria					Maandlike 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 wat die leerder probeer doen en watter aspekte opgelos moet word; Duidelike stelling van wat die doel van die programatuur sal wees	Die stelling is duidelik, maar het geringe tekortkominge.	Die stelling is vaag en 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				
	Huidige stelsels/prosesse							2	
	2	1	0						
Duidelike beskrywing van huidige stelsel/prosesse, insluitend beperkinge en uitdagings en hoe die program dit kan aanspreek	Geringe tekortkominge; Beskrywing nie altyd duidelik nie	Geen beskrywing van huidige stelsels/prosesse of beperkinge en uitdagings nie							
Onderzoek	Sleutelareas met betrekking tot die probleem					4			
	4	3	2	1	0				
	Onderzoek definieer/ verduidelik omvattend en duidelik die sleutelareas rakende die onderwerp; Toon goeie insig en begrip van al die sleutelareas van die onderwerp	Geringe tekortkominge; Toon insig in die 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; In aanvaarbare formaat	Minder omvattend – geringe tekortkominge	Geen verwysings nie							

Vereistes	Toevoervereistes				3	
	3	2	1	0		
	Omvattende lys van toevoervereistes verkry van gebruikers, goed gedefinieerd en verkry deur 'n geskikte metode, soos vraelyste, onderhoude, te gebruik	Geringe tekortkominge; Minder omvattende lys van toevoervereistes verkry van gebruikers; Nie so duidelik gedefinieerd nie, maar nog steeds aanvaarbaar	Min toevoervereistes verkry van verbruikers of vereistes is opgemaak deur die leerder; Vaag en nie duidelik gedefinieerd nie	Geen toevoervereistes verkry nie		
	Verwerkingsvereistes					
	3	2	1	0		
	Omvattende lys van verwerkingvereistes verkry van gebruikers of deur die leerder se navorsing, goed gedefinieerd en verkry deur 'n geskikte metode te gebruik	Minder omvattende lys van verwerkingvereistes verkry van gebruikers wat ten minste een metode gebruik; 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		
	Afvoervereistes					
	3	2	1	0		
	Omvattende lys van afvoervereistes verkry van gebruikers, goed gedefinieerd en verkry deur 'n geskikte metode, soos 'n vraelys, onderhoud, 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		
	Moontlike oplossing	Moontlike oplossing				4
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	Basiese beskrywing is gegee, maar sommige aspekte van die voorgestelde oplossing is vaag	Die beskrywing is vaag; Geen duidelike oplossing geïdentifiseer in die beskrywing nie	Geen moontlike oplossing gegee nie	
Die reikwydte (grense en aannames/eienskappe wat ingesluit sal word)						
3		2	1	0		
Die omvang 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			
Algemeen	Toepaslikheid van voorgestelde oplossing in die konteks van die scenario				2	
	2	1	0			
	Die meeste toepaslik; Goeie toepassing vir die scenario	Toepaslik; Toepassing in scenario nie altyd oortuigend nie	Nie toepaslik nie; Toepassing in scenario nie oortuigend nie			
Totaal:				30		

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 duidelik beskryf in terme van watter toevoer vereis word asook die formaat (bv. datumformaat) van die toevoer	Beskryf met geringe tekortkominge in terme van wat vereis word en in watter formaat	Beskrywing van toevoer vaag of onvolledig	Slegs gelys – geen beskrywing nie		
	Verwerking					
	3	2	1	0	3	
	Beskryf duidelik hoe data verwerk/ gemanipuleer sal word in terme van vereistes, formaat, berekeninge, formules, ens.; Kort, duidelike en korrekte beskrywing in alle gevalle waar toepaslik; Gebruiker het duidelikheid oor resultaat	Beskrywing van verwerking/ manipulering van data is gedoen/duidelik in die meeste 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; Nie gedoen in alle gevalle nie; Beskrywing is vaag/ onvolledig; Nie korrek/toepaslik in die meeste gevalle nie	Verwerking/Manipulering van data nie beskryf nie		
	Afvoer					
	3	2	1	0		
	Vereiste afvoer (skerm, verslae) 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 die meeste van die vereiste afvoer is nie geïdentifiseer nie of die formaat van afvoer is nie beskryf nie	Afvoer nie geïdentifiseer en beskryf nie	3	
	Datavalidering					
	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/-foute en in detail beskryf/aangetoon	'n Paar foutboodskappe aangedui en/of beskrywing van foutboodskappe is vaag	Geen foutboodskap aangedui vir enige datavalidering of fout 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 aangedui 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 soos validering en maskers nie aangedui nie			
	Tabelle				3					
	3		2				1		0	
Velde goed gegropeer 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 gegropeer in tabelle met geringe tekortkominge; Geringe herhaling van velde in tabelle; Ten minste twee tabelle met korrekte verwantskappe tussen tabelle; Toon korrekte primêre en vreemde sleutels		Velde meestal goed gegropeer in tabelle; Uitgebreide herhaling van velde in tabelle; Ten minste twee tabelle maar geen/verkeerde verwantskap tussen tabelle; Toon geen/verkeerde primêre en vreemde sleutels			Slegs een tabel/slegs velde gelys			
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	
	Gebruik abstrakte datatipes; 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; Nie altyd van toepassing nie; Hele program nie objek-georiënteerd 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 objek-georiënteerd		Geen abstrakte datatipes nie (klasse en objekte)	
'GUI'-ontwerp	Toevoer				3					
	3		2				1		0	
	Toepaslike toevoerkomponente om akkurate/geldige toevoer in alle gevalle waar vereis, te verseker; Gemotiveerde keuses waar vereis word		Toepaslike komponente waar vereis; Keuses meestal gemotiveer; Geringe tekortkominge				Toepaslik in die meeste gevalle waar vereis; Keuses nie altyd gemotiveer nie		Nie gedoen nie/nie toepaslike toevoerkomponente geïdentifiseer nie	
	Afvoer				3					
	3		2				1		0	
Toepaslike afvoerkomponente geïdentifiseer in alle gevalle waar vereis		Toepaslike afvoerkomponente geïdentifiseer waar vereis – geringe tekortkominge		Toepaslik in die meeste gevalle waar vereis; Keuses nie altyd gemotiveer nie			Nie gedoen nie/nie toepaslike afvoerkomponente geïdentifiseer nie			
Algemeen	Spreek die vereistes gespesifiseer in Fase 1 aan				2					
	2		1				0			
	Aandag gegee aan alle vereistes		Ten minste 50%				Minder as 50%			
Totaal:					30					

Assesering vir Fase 3:

Naam van leerder: _____

LET WEL: Fase 3 sal slegs geassesseer word wanneer Fase 1 and 2 voltooi is EN Fase 1 en 2 verband hou met die onderwerp gedek in Fase 3.

Kodering en Implementering: Kriteria					Moontlike Punt	Punt Behaal	
Databasis	Databasistabelle				4		
	4	3	2	1			0
	Getal tabelle voorsien 'n effektiewe oplossing – tabelle is genormaliseer (toepaslike getal tabelle om 'n doeltreffende oplossing te bied); Alle verwantskappe is goed gedefinieer deur korrekte primêre en vreemde sleutels te gebruik; Primêre en vreemde sleutels in verwante tabelle effektief en toepaslik	Getal tabelle bied 'n goeie oplossing; Ten minste een doeltreffende en toepaslike verwantskap wat die korrekte primêre en vreemde sleutels gebruik; Sommige primêre of vreemde sleutels in verwante tabelle nie toepaslik nie	Getal tabelle bied 'n gedeeltelike oplossing – tabelle nie genormaliseer nie (kon meer tabelle gebruik het vir 'n meer effektiewe oplossing); Die meeste van die primêre of vreemde sleutels nie geskik nie of slegs versteksleutels gebruik	Getal tabelle nie relevant tot 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 in elke databasis in plaas van een databasis)			Geen databasis gebruik nie
	Veldtipes						
	2	1	0				
	Alle velde met toepaslike datatipes	Datatipes van sommige velde nie toepaslik nie	Slegs verstektipes gebruik of geen databasis gebruik nie	2			
	Veldgroottes – databasis						
	2	1	0				
	Alle velde met toepaslike veldgroottes	Sommige van die velde is te groot/ klein	Slegs verstek-veldgroottes gebruik of geen databasis gebruik nie	2			
	Beskrywende veldname in databasis				1		
LET WEL: Assesseer die kode indien die program nie uitvoer nie. Hierdie afdeling het te make met die programmering in Delphi/Java, uitgesluit SQL.							
Programmeringsoplossing	Gebruikersgedefinieerde datastrukture (uitgesluit databasistabelle)				3		
	3	2	1	0			
	Toepaslike en mees doeltreffende datastrukture gebruik om die probleem in alle gevalle op te los	Toepaslike en mees doeltreffende datastrukture in die meeste gevalle gebruik of met geringe tekortkominge	Toepaslike en mees doeltreffende gebruik van datastrukture in minder as 50% van gevalle	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	Geringe tekortkominge	Nie betekenisvol nie				
Seleksiestrukture				3			
3	2	1	0				
Toepaslike en mees doeltreffende keusestrukture in alle gevalle gebruik om die probleem op te los	Toepaslike en mees doeltreffende keusestrukture in die meeste gevalle gebruik	Onvanpaste of ondoeltreffende gebruik van keusestrukture in die meeste gevalle	Nie gebruik nie				

Herhalingstrukture							
3		2		1		0	
Toepaslike en die mees doeltreffende herhalingstrukture in alle gevalle gebruik om die probleem op te los		Toepaslike en die mees doeltreffende herhalingstrukture in die meeste gevalle gebruik		Ontoepaslike of ondoeltreffende gebruik van herhalingstrukture in die meeste gevalle		Nie gebruik nie	
						3	
Komplekse programmeringstegnieke (bv. speel videosnitte, geleende kode, rafels ('threads'), netwerk)							
3		2		1		0	
Dit werk korrek; Toepaslik gebruik en voeg waarde toe tot die oplossing		Dit werk korrek; Nie altyd toepaslik gebruik nie of voeg nie regtig waarde toe tot die oplossing 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	
						3	
Datavloei en prosesse (gebruikersgedefinieerde parameteroordrag)							
3		2		1		0	
Uitstekende interaksie/kommunikasie tussen modules/klasse; Sluit gevorderde datatipes as parameters of terugstuurtipies in		Vaardige/voldoende/'n mate van kommunikasie tussen modules/klasse met klein foute; Sluit 'n 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/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	
						3	
Oplossingalgoritmes							
3		2		1		0	
Alle oplossingalgoritmes gebruik om probleem op te los is toepaslik en doeltreffend		Toepaslik en doeltreffend met geringe tekortkominge		Meestal ontoepaslik of nie doeltreffend nie		Totaal ontoepaslike oplossingalgoritmes of meestal ondoeltreffend	
						3	
Korrektheid van oplossingalgoritmes							
3		2		1		0	
Geen logiese foute nie; Alle verwerkingsresultate korrek		Geringe tekortkominge; Baie min logiese foute; Baie min van die resultate nie korrek nie		Logiese foute; Baie resultate foutief		Baie logiese foute; Byna al die resultate is foutief/min van die vereiste resultate word gelewer	
						3	
Toevoerstrategieë							
3		2		1		0	
Die mees toepaslike, doeltreffende toevoerstrategieë in alle gevalle gebruik, bv. toevoer van tekslêers, databasistabelle, gebruikerstoevoer		Toepaslik en doeltreffend met geringe tekortkominge		Sommige strategieë kon meer geskik/doeltreffend wees		Meestal ontoepaslik of ondoeltreffend	
						3	

NSS – Assesseringsinstrumente

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

Grafiese gebruikerskoppelvlak ('GUI')	Verskillende skerms toepaslik gebruik (vensters/paneel/'tab sheets', ens.)			1
	Komponente vir toevoer/afvoer gebruik (geïmplementeer soos vir Fase 2) OF toepaslike veranderinge gemaak vanaf Fase 2 se terugvoer)			2
	2	1	0	
	Altyd toepaslik, die doeltreffendste	Geringe tekortkominge	Meestal nie toepaslik nie/ ondoeltreffend	
	Byskrifte/lokboodskappe met eksakte formate vir toevoer			2
	2	1	0	
	Konstant toegepas regdeur die hele projek waar vereis word	Toegepas in die meeste gevalle waar vereis word	Meestal nie toegepas nie	
	Konsekwente uitleg (dieselfde voorkoms en gevoel regdeur)			2
	2	1	0	
	Vir alle skerms (dieselfde kleure, skrifstipes gebruik regdeur program)	Die meeste van die skerms (sommige skerms verskillende kleure, skrifstipes, ens.) – geringe tekortkominge	Uitleg meestal nie konsekwent nie	
	Insiggewende afvoer/verslae 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 skrifgrootte, uitleg, kleur, ens.)	Die 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 gegroepeer, bv. adresinligting, vir alle skerms	Tipe toevoer/afvoer meestal saam gegroepeer – geringe tekortkominge	In die meeste gevalle nie saam gegroepeer 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		
Konteks sensitief			2	
2	1	0		
Konteks-sensitiewe hulp beskikbaar ('tool tip'-teks ingesluit), toepaslik en effektief	Nie altyd toepaslik of effektief nie	Nie beskikbaar nie		
Ontwerp vs. teikengebruiker			2	
2	1	0		
Ontwerp neem teikengehoor in ag (ouderdom, vlak van geletterdheid, beperkte visie, toepaslikheid van beelde, ens.); Ontwerp geskik vir teikengebruiker	In ag geneem, maar geringe tekortkominge	Heeltemal ontoepaslik		

Dokumentering	Interne dokumentering				2						
	2		1				0				
	Kode regdeur program is 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				
Dokumentering	Skeiding van afdelings				2						
	2		1				0				
	Afdelings in die kode van die program duidelik geskei om leesbaarheid te verhoog (spasiëring, kommentaar vir metode/onderafdeling, ens.)		Sommige afdelings geskei				Nie geskei nie				
Algemene evaluering	Tydbestuur				3						
	3		2				1		0		
	Alle spertye gehaal – al 3 fases en al die vereiste werk is gedoen		2 spertye gehaal of lewer betyds in, maar sommige werk is nie gedoen nie				Ten minste een sperdatum gehaal of betyds ingelewer, maar die meeste werk nie gedoen nie		Altyd laat, nooit gedoen nie		
	Vermoë om kode wat lukraak gekies is, se doel en werking te verduidelik				4						
	4		3				2		1		0
Verduidelik alle gekose kode duidelik en met selfvertroue; Toon uitstekende insig		Verduidelik gekose kode met geringe tekortkominge; Toon insig		Verduidelik sommige van die gekose kode; Toon 'n bietjie insig in sommige van die kode			Nie in staat om die meeste van die gekose kode toepaslik te verduidelik nie; Gebrek aan insig		Nie in staat om te verduidelik nie		
Algemene evaluering	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 'n bietjie aanpassing.				Sommige dele kan in die werklike situasie gebruik word.		Totaal irrelevant; Sal nie in 'n werklike situasie werk nie		
Totaal:					110						