



# basic education

Department:  
Basic Education  
**REPUBLIC OF SOUTH AFRICA**

**NASIONALE  
SENIOR SERTIFIKAAT**

**GRAAD 12**

**INFT.1**

**INLIGTINGSTEGNOLOGIE V1**

**FEBRUARIE/MAART 2017**

**PUNTE: 150**

**TYD: 3 uur**

Hierdie vraestel bestaan uit 16 bladsye.

# OGGENDSESSIE



**INSTRUKSIES EN INLIGTING**

1. Hierdie vraestel is in DRIE afdelings ingedeel. Kandidate moet AL DRIE afdelings beantwoord.
2. Die duur van hierdie eksamen is drie uur. As gevolg van die aard van hierdie eksamen is dit belangrik om daarop te let dat jy nie toegelaat sal word om die eksamenlokaal voor die einde van die eksamensessie te verlaat nie.
3. Hierdie vraestel is opgestel met programmeringsterme wat nie programmeringstaal-spesifiek is nie (Delphi/Java (deur die Netbeans IDE te gebruik)).
4. Maak seker dat jy die vrae beantwoord volgens die spesifikasies wat in elke vraag gegee word. Punte sal volgens die voorgeskrewe vereistes toegeken word.
5. Beantwoord slegs wat in elke vraag gevra word. Byvoorbeeld, indien die vraag nie datavalidering vra nie, sal geen punte vir datavalidering toegeken word nie.
6. Jou programme moet op só 'n manier gekodeer word dat dit met enige data sal werk en nie net met die voorbeelddata wat voorsien is of enige data-uittreksels wat in die vraestel verskyn nie.
7. Roetines, soos soek, sorteer en seleksie, moet vanuit eerste beginsels ontwikkel word. Jy mag NIE die ingeboude funksies van 'n programmeringstaal vir enige van hierdie roetines gebruik NIE.
8. Alle datastrukture moet deur jou, die programmeerder, verklaar word, tensy die datastrukture voorsien word.
9. Jy moet jou werk gereeld stoor op die disket/CD/DVD/geheuestokkie wat aan jou gegee is, of op die skyfspasie wat vir hierdie eksamensessie aan jou toegeken is.
10. Maak seker dat jou eksamenommer as kommentaar verskyn in elke program wat jy kodeer, asook op elke gebeurtenis wat aangedui word.
11. Indien dit vereis word, druk die programmeringskode van al die programme/klasse wat jy voltooi het. Jy sal ná die eksamensessie 'n halfuur tyd vir drukwerk gegee word.
12. Aan die einde van hierdie eksamensessie moet jy 'n disket/CD/DVD/geheuestokkie inlewer met al jou werk daarop gestoor OF jy moet seker maak dat al jou werk op die skyfspasie gestoor is wat vir hierdie eksamensessie aan jou toegeken is. Maak seker dat al die lêers gelees kan word.



13. Die lêers wat jy benodig om hierdie vraestel te voltooi, is aan jou gegee op die disket/CD/DVD/geheuestokkie of op die skyfspasie wat aan jou toegeken is. Die lêers word in die vorm van wagwoordbeskernde, uitvoerbare lêers verskaf.

**LET WEL:**

- Delphi-kandidate moet die lêer **DelphiDataAFRMaart2017.exe** gebruik.
- Java-kandidate moet die lêer **JavaDataAFRMaart2017.exe** gebruik.

Doen die volgende:

- Dubbelklik op die wagwoordbeskernde uitvoerbare lêer.
- Klik op die 'Extract'-knoppie.
- Sleutel die volgende wagwoord in: **Prof4U@#!**

Nadat dit onttrek ('extracted') is, sal die volgende lys met lêers in die lêergids **DelphiDataAFRMaart2017-/JavaDataAFRMaart2017** beskikbaar wees:

**DELPHI-LÊERS****Vraag1:**

Tipe1.PNG  
Tipe2.PNG  
Vraag1\_P.dpr  
Vraag1\_P.dproj  
Vraag1\_P.res  
Vraag1\_U.dfm  
Vraag1\_U.pas

**Vraag2:**

DataJanuarie2017.txt  
KaartHouer\_U.pas  
Vraag2\_P.dpr  
Vraag2\_P.dproj  
Vraag2\_P.res  
Vraag2\_U.dfm  
Vraag2\_U.pas

**Vraag3:**

Vraag3\_P.dpr  
Vraag3\_P.dproj  
Vraag3\_P.res  
Vraag3\_U.dfm  
Vraag3\_U.pas

**JAVA (NETBEANS)-LÊERS****Vraag1:**

Tipe1.PNG  
Tipe2.PNG  
Vraag1.form  
Vraag1.java

**Vraag2:**

DataJanuarie2017.txt  
KaartHouer.java  
Vraag2.form  
Vraag2.java

**Vraag3:**

Vraag3.form  
Vraag3.java



**SCENARIO**

Lojaliteitskaarte word dikwels gebruik om klanteprofiele te skep. Dit is ook 'n gewilde manier om klante te motiveer om winkels te besoek deur gereeld lojaliteitspunte toe te ken vir aankope wat gedoen word.

**AFDELING A****VRAAG 1: ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

'n Aansoeker moet 'n aansoekvorm voltooi om 'n lojaliteitskaart te kry.

Doen die volgende:

- Kompileer en voer die program uit wat in die **Vraag1**-lêergids voorkom. Die gebruikerskoppelvlak vertoon vier verskillende afdelings wat VRAAG 1\_1 tot VRAAG 1\_4 genoem word. Die program het tans geen funksionaliteit nie.

Voorbeeld van grafiese gebruikerskoppelvlak (GGK) ('GUI'):

**Aansoek vir 'n lojaliteitskaart**

<p><b>Vraag 1_1</b></p> <p>Naam en van <input style="width: 150px;" type="text"/></p> <p>ID-nommer <input style="width: 150px;" type="text"/> *</p> <p style="text-align: center; margin-top: 10px;"><input type="button" value="Vraag 1_1"/></p>	<p><b>Vraag 1_3</b></p> <p>Sleutel jou finale aankoopbedrag vir vandag in <input style="width: 100px;" type="text"/></p> <p style="text-align: center; margin-top: 5px;"><input type="button" value="Vraag 1_3"/></p> <p>Voordeel</p> <p><input style="width: 150px;" type="text"/></p>
<p><b>Vraag 1_2</b></p> <p><b>Selekteer die tipe lojaliteitskaart wat jy verkies</b></p> <p>Onmiddellike afslag      Donasie vir liefdadigheid</p> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> </div>	<p><b>Vraag 1_4</b></p> <p><input type="checkbox"/> Besigheidskaart</p> <p style="text-align: center; margin-top: 10px;"><input type="button" value="Vraag 1_4"/></p> <p>Jou lojaliteitskaartnummer</p> <p><input style="width: 30px;" type="text"/> - <input style="width: 30px;" type="text"/> - <input style="width: 30px;" type="text"/></p>

- Voltooi die kode vir elke afdeling van VRAAG 1, soos in VRAAG 1.1 tot VRAAG 1.4 op die volgende bladsy beskryf word.

## 1.1 Knoppie [Vraag 1\_1]

'n Byskrif ('label') met die naam **IbiWelkomBoodskap** is geskep en gestel om nie sigbaar te wees nie. Skryf kode om die inligting wat deur die gebruiker ingesleutel word, soos volg te valideer:

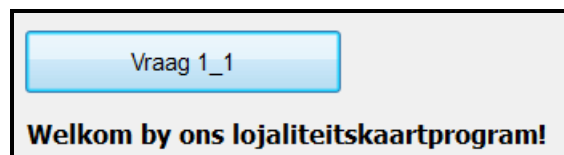
- Naam en van

Die toevoer moet ten minste een spasie bevat om geldig te wees. As die toevoer nie geldig is nie, vertoon 'n boodskap om die spesifieke fout aan te dui deur 'n dialoogblokkie ('dialog box') te gebruik.

- ID-nommer

Die asterisk-karakter (\*) langs die teksblokkie dui aan dat die veld nie oop gelaat kan word nie. 'n ID-nommer moet uit 13 syfers bestaan om geldig te wees. As die ID-nommer-veld nie enige inligting bevat nie of die ingevulde ID-nommer nie uit 13 syfers bestaan nie, vertoon 'n boodskap om aan te dui watter spesifieke fout voorgekom het deur 'n dialoogblokkie te gebruik.

As die inhoud in beide velde geldig is, moet die byskrif wat die verwelkomingsboodskap vertoon, op sigbaar gestel word.



Voorbeeld van boodskap as geen spasie in die naam-en-van-veld ingesleutel is nie:

Die naam-en-van-veld bevat nie 'n spasie nie.

Voorbeeld van boodskap as die ID-nommer-veld oop gelaat is:

'n ID-nommer moet ingesleutel word.

Voorbeeld van boodskap as die ID-nommer wat ingesleutel is, nie uit 13 syfers bestaan nie:

Die ID-nommer bestaan nie uit 13 syfers nie.

(14)

## 1.2 Knoppies met prentjies [Vraag 1\_2]

'n Klant moet die tipe lojaliteitskaart wat hy/sy verkies, selekteer. Die prentjies op die knoppies in hierdie afdeling van die GGK wys twee tipes lojaliteitskaarte. Wanneer 'n spesifieke prentjie geselekteer word (knoppie geklik), moet die program die volgende doen:

- Die opskrif in die byskrif ('label') bokant die geselekteerde prentjie moet na vetdruk ('bold') verander word.



- 'n Kode vir die tipe lojaliteitskaart wat geselekteer is, moet aan die globale veranderlike **tipeKaart** wat verskaf is, toegeken word. Die kode AF verteenwoordig onmiddellike afslag en die kode LI verteenwoordig 'n donasie vir liefdadigheid.

**LET WEL:** Slegs die opskrif van die knoppie wat tans geselekteer is, moet in vetdruk ('bold') verskyn. As die gebruiker die ander tipe lojaliteitskaart selekteer, moet die opskrif van die lojaliteitskaart wat voorheen geselekteer is, NIE in vetdruk ('bold') verskyn NIE. (6)

### 1.3 Knoppie [Vraag 1\_3]

Die finale aankoopbedrag vir vandag moet in die teksblokkie ('text box') ingesleutel word.

Die program moet die volgende voordeel toestaan aan die houer van die tipe lojaliteitskaart wat in VRAAG 1\_2 geselekteer is:

- Onmiddellike afslag (AF):  
1,5% afslag moet op die aankoopbedrag toegestaan word.
- Donasie vir liefdadigheid (LI):  
'n Bedrag van R3 vir elke R100 of deel van R100 wat deur die lojaliteitskaarthouer gespandeer is, sal aan 'n liefdadigheidsorganisasie geskenk word.

Vertoon 'n boodskap wat die voordeel wat toegestaan is, afhangend van die tipe lojaliteitskaart, sal beskryf.

Voorbeeld van afvoer as die kode van die lojaliteitskaart AF en die aankoopbedrag R1 200 is:

Sleutel jou finale aankoopbedrag vir vandag in 1200

Vraag 1\_3

Voordeel

1.5% afslag op jou aankoopbedrag is R18.00.

**LET WEL:** Die afslagbedrag moet as 'n geldeenheid met twee desimale syfers vertoon word.

Voorbeeld van afvoer as die kode van die lojaliteitskaart LI en die aankoopbedrag R230 is:

Sleutel jou finale aankoopbedrag vir vandag in 230

Vraag 1\_3

Voordeel

'n Bedrag van R9.00 sal aan liefdadigheid geskenk word.

(15)



## 1.4 Knoppie [Vraag 1\_4]

Die klant se ID-nommer wat in VRAAG 1\_1 ingesleutel is en die tipe lojaliteitskaart wat in VRAAG 1\_2 geselekteer is, word benodig om die klant se lojaliteitskaartnommer saam te stel. 'n Lojaliteitskaartnommer bestaan uit drie dele. Elke deel van die nommer moet soos volg saamgestel en in die komponente wat verskaf is, vertoon word:

### Deel 1:

Die eerste deel van die lojaliteitskaartnommer moet óf uit die letter B bestaan as die besigheidskarta-kontroleblokkie ('check box') geselekteer is, óf die letter P as die besigheidskarta-kontroleblokkie ('check box') nie geselekteer is nie, gevolg deur die eerste drie syfers van die klant se ID-nommer.

### Deel 2:

'n String met die naam **karakString**, wat die volgende 10 karakters bevat, is voorsien as deel van die kode wat verskaf is:

! @ # \$ % ] & \* ^ ~

Elke karakter in die string verteenwoordig 'n syfer (0–9). Dit beteken dat die eerste karakter (!) die syfer 0 verteenwoordig, die tweede karakter (@) die syfer 1, en so aan.

Hierdie deel van die lojaliteitskaartnommer moet bestaan uit die laaste syfer van die klant se ID-nommer, gevolg deur die karakter uit **karakString** wat daardie syfer verteenwoordig.

Voorbeeld van hierdie deel van die lojaliteitsnommer as die klant se ID-nommer 8802128756084 is:

4%

### Deel 3:

Hierdie deel van die lojaliteitskaartnommer bestaan uit die kode wat die tipe lojaliteitskaart (AF of LI) verteenwoordig en in VRAAG 1\_2 bepaal is.

Voorbeeld van die lojaliteitskaartnommer van 'n klant met die volgende profiel:

- Aansoek vir 'n besigheid
- ID-nommer: 9210140012083
- Tipe kaart: AF

Jou lojaliteitskaartnommer		
B921	- 3\$	- AF

(13)

- Sleutel jou eksamennommer as kommentaar in die eerste reël van die programlêer in.
- Stoor jou program.
- Maak 'n drukstuk van die kode as dit vereis word.

**TOTAAL AFDELING A: 48**



**AFDELING B**

**VRAAG 2: OBJEK-GEÖRIENTEERDE PROGRAMMERING**

Die plaaslike supermark het 'n lojaliteitskaartstelsel waar klante belonings kan ontvang, afhangend van die aantal besoeke, die aantal lojaliteitspunte wat versamel is en die persentasie gesondheidskos wat aangekoop is.

Doen die volgende:

Delphi-programmeerders	Java-programmeerders
<ul style="list-style-type: none"> <li>Maak die onvoltooide program in die <b>Vraag2-lêergids</b> oop.</li> <li>Maak die onvoltooide objekklas <b>KaartHouer_U.pas</b> oop.</li> <li>Voeg jou eksamennummer as kommentaar in die eerste reël van beide lêer <b>Vraag2_U.pas</b> en <b>KaartHouer_U.pas</b> in.</li> </ul>	<ul style="list-style-type: none"> <li>Maak die onvoltooide program in die <b>Vraag2-lêergids</b> oop.</li> <li>Maak die onvoltooide objekklas <b>KaartHouer.java</b> oop.</li> <li>Voeg jou eksamennummer as kommentaar in die eerste reël van beide objekklas <b>Vraag2.java</b> en <b>KaartHouer.java</b> in.</li> </ul>

- Kompileer en voer die program uit. Die program het tans geen funksionaliteit nie.

Voorbeeld van grafiese gebruikerskoppelvlak (GGK):

### Lojaliteitsbeloningstelsel

Vraag 2\_2\_1

Lojaliteitskaartnummer

B921-3\$-AF ▾

Selfoonnummer            0812345678

Lojaliteitskaartpunte    2130

Toegangskode

2.2.1 - Kontroleer toegangskode

Vraag 2\_2\_2

2.2.2 - Vertoon kaarthouer se besonderhede

- Voltooi die kode vir hierdie program, soos in VRAAG 2.1 en VRAAG 2.2 hieronder gespesifiseer.

2.1 Die gegewe onvolledige objekklas (**TKaartHouer/KaartHouer**) bevat die verklaring van vyf attribute, die **toString()**-metode en 'n onvolledige **isReg()**-metode.





Die gegewe attribute van die **KaartHouer**-objek:

NAME VAN ATTRIBUTE		BESKRYWING
Delphi	Java	
fKaartNommer	kaartNommer	'n Unieke nommer vir die kaarthouer
fSelNommer	selNommer	Die selfoonnommer van die kaarthouer
fAantBesoeke	aantBesoeke	Die aantal kere wat die kaarthouer die winkel gedurende die maand besoek het
fLojaliteitsPunte	lojaliteitsPunte	Die totale aantal lojaliteitspunte wat tot nou toe op hierdie kaart ontvang is
fGesondVlak	gesondVlak	'n Karakter wat die gesondheidsvlak van die kaarthouer aandui op grond van die hoeveelheid gesondheidskos wat gedurende die maand aangekoop is: S – Silwer (Laagste vlak) G – Goud P – Platinum (Hoogste vlak)

Voltooi die kode in die gegewe **KaartHouer**-klas (**TKaartHouer/KaartHouer**) soos wat in VRAAG 2.1.1 tot VRAAG 2.1.6 hieronder beskryf word.

2.1.1 Skryf kode vir 'n **konstruktor**-metode wat die kaartnommer, selfoonnommer en lojaliteitskaartpunte wat reeds ontvang is, as parameterwaardes sal ontvang. Ken hierdie waardes aan die relevante attribute toe.

Die laagste kode vir die vlak gesondheidskos (S) moet aan die gesondheidsvlak-attribuut toegeken word. Die aantal besoeke se attribuut moet op 0 gestel word. (5)

2.1.2 Skryf 'n **mutator**-metode om die getal besoeke se attribuut op 'n waarde wat as 'n parameter ontvang is, te kan stel. (2)

2.1.3 Skryf kode vir 'n metode met die naam **vermeerderLojaliteitsPunte** wat die totale bedrag wat hierdie maand spandeer is, as 'n parameter ontvang en die huidige aantal lojaliteitspunte sal vermeerder volgens die bedrag wat ontvang is.

Gebruik die volgende kriteria:

Vir elke vier rand (R4) wat gespandeer is, word een lojaliteitspunt toegeken. (5)

2.1.4 Skryf kode vir 'n metode met die naam **opdateerGesondVlak** wat die totale bedrag wat hierdie maand gespandeer is en die bedrag wat aan gesondheidskos spandeer is, as parameters sal ontvang. Bereken die persentasie wat aan gesondheidskos spandeer is en stel die gesondheidsvlak-attribuut volgens die kriteria wat in die tabel hieronder verskaf is.

Kode van die vlak	Persentasie-aanduiding
S	Minder as 10%
G	10% of meer, maar minder as 40%
P	40% of meer

(7)



- 2.1.5 'n Toegangskode is nodig om 'n klant se lojaliteitspunte op te dateer. Die klant se selfoonnommer is gebruik om die toegangskode saam te stel.

Skryf kode om die **isReg**-metode wat verskaf is, te voltooi. Die metode moet toets of die toegangskode, wat as 'n parameter ontvang is, korrek is deur die volgende instruksies op die selfoonnommer-attribuut toe te pas:

- Verwyder al die nulle (0) uit die selfoonnommer.
- Bepaal die som van die oorblywende syfers soos volg:
  - As die getal oorblywende syfers ewe is, moet elke twee opeenvolgende syfers wat aan die linkerkant begin, as 'n tweesyferwaarde beskou word. Tel al hierdie tweesyferwaardes bymekaar.

Voorbeeld:

As die oorspronklike selfoonnommer 082 340 2113 was, sal die oorblywende syfers sonder nulle 82342113 wees. Die som van die tweesyferwaardes is:

$$82 + 34 + 21 + 13 = 150$$

- As die getal oorblywende syfers onewe is, moet die eerste syfer as 'n eensyferwaarde beskou word en daarna moet elke twee opeenvolgende syfers as 'n tweesyferwaarde beskou word. Tel al hierdie waardes bymekaar.

Voorbeeld:

As die oorspronklike selfoonnommer 082 034 2101 was, sal die oorblywende syfers sonder nulle 8234211 wees. Die som van die waardes is:

$$8 + 23 + 42 + 11 = 84$$

(13)

- 2.1.6 Skryf kode vir 'n metode met die naam **identifiseerSterKoper** wat 'n string sal terugstuur wat aandui of die kaarthouer 'n STER-koper is, of nie.

'n Kaarthouer is 'n STER-koper as die volgende van toepassing is:

- Die kaarthouer het meer as 2 000 lojaliteitspunte verdien en die winkel is gedurende die maand meer as 10 keer besoek.  
OF
- Die gesondheidsvlak van die kaarthouer is platinum.

Die string wat teruggestuur word, moet of 'STER-koper' of 'n leë string wees.

(6)



2.2 Inligting oor die besoeke en aankope van kaarthouers word slegs vir een maand in 'n tekslêer gestoor. Die tekslêer met die naam **DataJanuarie2017.txt** bevat die transaksies wat gedurende Januarie deur vier van die kaarthouers gedoen is.

Die inhoud van die tekslêer is soos volg:

Elke drie reëls verteenwoordig 'n transaksie vir 'n spesifieke lojaliteitskaarthouer:

Reël 1: Die kaartnommer van die kaarthouer, byvoorbeeld B921-3\$-DI

Reël 2: Die totale bedrag van die aankope vir hierdie besoek, byvoorbeeld 1459.30

Reël 3: Die bedrag wat op gesondheidskos spandeer is, byvoorbeeld 72.30

Voorbeeld van die eerste 12 reëls van die **DataJanuarie2017.txt**-tekslêer:

```
B921-3$-AF
1459.30
72.30
B870-4%-LI
899.80
22.50
P730-7*-AF
299.30
62.85
B870-4%-LI
869.30
182.55
```

Doen die volgende om die kode vir elke knoppie in die hoofvormeenheid (Delphi)/GGK-klas (Java) soos in die gedeelte wat volg, te voltooi.

**LET WEL:** Die objek **objKaartHouer** is globaal verklaar as deel van die kode wat verskaf is.

### 2.2.1 Knoppie – [2.2.1 – Kontroleer toegangskode]

Die gebruiker moet 'n kaartnommer selekteer uit die vier kaartnommers wat in die kombinasieblokkie ('combo box') verskaf is.

Sodra 'n kaartnommer in die kombinasieblokkie geselekteer is, word die kaarthouer se selfoonnommer en die aantal lojaliteitspunte wat reeds uit vorige aankope verdien is, vertoon met die kode wat verskaf is.

Die gebruiker moet die korrekte toegangskode vir die kaartnommer wat geselekteer is, insleutel sodat die program die kaarthouer se data, wat in die tekslêer gestoor is, kan verwerk.

Skryf kode om die volgende te doen:

- Gebruik die kaarthouer se kaartnommer, selfoonnommer en lojaliteitspunte van die kaarthouer om 'n kaarthouer-objek te instansieer.



- Gebruik die metode **isReg** om te bepaal of die toegangskode wat vir hierdie kaarthouer ingesleutel is, die korrekte toegangskode is, of nie.

As die korrekte toegangskode ingesleutel is, gebruik die data in die **DataJanuarie2017.txt**-tekslêer om die volgende met betrekking tot die kaarthouer se aankope te bepaal:

- Die totale bedrag wat vir die maand spandeer is
- Die bedrag wat aan gesondheidsverwante produkte spandeer is
- Die aantal kere wat die gebruiker die winkel besoek en aankope gedoen het

Gebruik hierdie inligting en die relevante metodes uit die objekklas om die aantal lojaliteitspunte, die gesondheidsvlakkode en die aantal kere wat die kaarthouer die winkel vir aankope besoek het, op te dateer. Maak die knoppie vir VRAAG 2.2.2 aktief.

As die toegangskode NIE korrek is NIE, vertoon 'n boodskap wat aandui dat die aanteken ('login') nie suksesvol was nie en maak die toegangskode-tekstblokkie skoon.

Die data in die tabel hieronder kan vir toetsdoeleindes gebruik word.

Kaartnommer	Toegangskode
B921-3\$-DI	188
B870-4%-CH	41
P730-7*-DI	338
B810-2#-DI	170

(24)

## 2.2.2 Knoppie – [2.2.2 – Vertoon kaarthouer se besonderhede]

Skryf kode om die **toString**-metode te gebruik om die data van die kaarthouer-objek in die afvoerarea wat verskaf is, te vertoon.

Die metode wat in VRAAG 2.1.6 geskep is, moet gebruik word om aan te dui of die kaarthouer 'n STER-koper is.

Voorbeeld van afvoer vir kaartnommer B921-3\$-AF:

```
B921-3$-AF
Kontaknommer: 0812345678

Opgedateerde getal lojaliteitspunte: 8190
Getal besoeke: 23
Gesondheidsevalueringstatus: S
STER-koper
```



Voorbeeld van afvoer vir kaartnommer B870-4%-LI:

B870-4%-LI Kontaknommer: 0822001100  Opgedateerde getal lojaliteitspunte: 6519 Getal besoeke: 5 Gesondheidsevalueringstatus: G
---

(3)

- Sleutel jou eksamennommer as kommentaar in die eerste reël van die klas en die vorm in.
- Stoor al die lêers.
- Druk die kode van die klas en die vorm (Delphi)/albei klasse (Java) indien dit vereis word.

**TOTAAL AFDELING B: 65**



**AFDELING C****VRAAG 3: PROBLEEMOPLOSSINGPROGRAMMERING****SCENARIO**

'n Sosialemedia-vriendskapstoep ('app') met die naam *Dine-Is-Fine* is spesiaal vir die 10 restaurante by jou plaaslike inkopiesentrum ontwerp. Jy en jou vriende is geregistreerde gebruikers van hierdie toep. Die toep kan gebruik word om die plek waar jy is, met ander geregistreerde gebruikers te deel.

Doen die volgende:

- Kompileer en voer die program in die **Vraag3**-lêergids uit. Die program het tans geen funksionaliteit nie.
- Voltooi die kode vir elke vraag, soos in VRAAG 3.1 tot VRAAG 3.3 beskryf is.

**GGK ('GUI') wat voorsien is:**

The screenshot shows a graphical user interface for an application titled "Dine-Is-Fine-toep". The interface is contained within a window with a grey title bar. On the left side, there is a vertical stack of controls: a button labeled "Vertoon uitleg", a text label "Selekteer restaurant(e) waar jou vriend(e) gaan eet" above a dropdown menu showing the letter "A", and another button labeled "Spoor naaste vriend(e) op". To the right of these controls is a large, empty rectangular area labeled "Afvoerarea". Below the main control area is a wide, empty horizontal space.

**Data wat voorsien is:**

Jy word voorsien van 'n tweedimensionele skikking met die naam **arrSentrum** wat gevul is met karakters wat die uitleg van die inkopiesentrum verteenwoordig. Die karakters A tot J verteenwoordig 10 restaurante wat jy en jou vriende gereeld besoek.



**Afvoerarea:**

Besluit op 'n geskikte afvoercomponent wat in die afvoerarea wat verskaf is, geplaas moet word om die inhoud van die tweedimensionele skikking te vertoon.

**LET WEL:**

- Jy mag NIE die data wat voorsien is, met die hand verander NIE. Kode moet geskryf word om die data wat voorsien is volgens die vereistes te manipuleer.
- Goeie programmeringstechnieke en modulêre ontwerp moet in die ontwerp en kodering van jou oplossing toegepas word.
- **GEEN punte sal vir kodering met vaste waardes toegeken word NIE.** Gebruik kontrolestrukture en veranderlikes waar nodig.

**3.1 Knoppie [Vertoon uitleg]**

Die program moet die uitleg van die inkopiesentrum vertoon. Skryf kode om die data wat in die tweedimensionele skikking voorsien is, te gebruik om die uitleg in netjiese kolomme te vertoon.

Voorbeeld van afvoer:

A	X	O	*	O	X	J
B	X	O	O	O	X	I
C	X	O	O	O	X	H
D	X	O	O	O	X	G
E	X	O	O	O	X	F

(5)

Verduideliking van die data in die skikking:

- Die karakters A tot J verteenwoordig die posisie van elk van die 10 restaurante in die inkopiesentrum.
- Die karakter X is 'n plekhouer wat aandui dat nie een van jou vriende tans in 'n restaurant is nie.
- Die karakter O dui die oop area aan waar jy kan rondbeweeg.
- Die asterisk-karakter (\*) dui jou posisie (eerste ry en vierde kolom) aan.

**LET WEL:** Vir die doel van hierdie vraag sal jy altyd in dieselfde posisie bly.

**3.2 Kombinasieblokkie ('Combo box') met restaurantkodes**

Jy word versoek om 'n restaurant uit die kombinasieblokkie te selekteer waar een van jou vriende gaan eet.

Skryf kode om die X-plekhouer by die geselekteerde restaurant met 'n hutsmerk-simbool (#) te vervang om aan te dui dat 'n vriend by die restaurant is en vertoon 'n opgedateerde uitleg van die inkopiesentrum.



Voorbeeld van afvoer as twee van jou vriende by restaurant C en F onderskeidelik eet:

A	X	O	*	O	X	J
B	X	O	O	O	X	I
C	#	O	O	O	X	H
D	X	O	O	O	X	G
E	X	O	O	O	#	F

(13)

**3.3 Knoppie – [Spoor naaste vriend(e) op]**

Die program moet bepaal by watter restaurant(e) in die inkopiesentrum jou vriende tans is en hoe ver hulle is van waar jy is. Elke vertikale of horisontale beweging op die matriks verteenwoordig 'n afstand van 4,5 meter. Vanaf die restaurant(e) waar jou vriende is, bepaal en vertoon die restaurant(e) wat die naaste aan jou is.

Voorbeeld van afvoer as jou vriende by restaurant C en F eet:

Restaurant C - ongeveer 22.5 meter Restaurant F - ongeveer 31.5 meter  Jou naaste vriend(e) is by: C
---

Voorbeeld van afvoer as jou vriende by restaurant C, F en H eet:

Restaurant C - ongeveer 22.5 meter Restaurant H - ongeveer 22.5 meter Restaurant F - ongeveer 31.5 meter  Jou naaste vriend(e) is by: C H
---

**LET WEL:** Jy kan nie diagonaal op die matriks beweeg nie.

**PUNTETOEKENNING VIR VRAAG 3.3**

Spoor vriende op	3
Bepaal die naam van die restaurant	4
Bereken die afstand	6
Vind die vriend(e) wat die naaste aan jou is	3
Afvoer om die posisie van vriend(e) en afstand aan te dui	2
Korrekte formaat van afvoer vir die vriend(e) wat die naaste aan jou is	1
<b>TOTAAL</b>	<b>19</b>

- Sleutel jou eksamennummer as kommentaar in die eerste reël van die programlêer in.
- Stoor jou program.
- Maak 'n drukstuk van die kode as dit vereis word.

**TOTAAL AFDELING C:  
GROOTTOTAAL:**

**37  
150**





Example of output if two of your friends are dining at restaurant C and F respectively:

A	X	O	*	O	X
B	X	O	O	O	X
C	#	O	O	O	X
D	X	O	O	O	X
E	X	O	O	O	X

3.3 Button [Locate nearest friend(s)]

The program must determine at which restaurant(s) in the mall your friends currently are and how far they are from where you are. Every vertical or horizontal movement on the matrix represents a distance of 4,5 metres. From the restaurant(s) where your friends are, determine and display the restaurant(s) closest to where you are.

Example of output if your friends are dining at restaurants C and F:

Restaurant C - approximately 22,5 metres  
 Restaurant F - approximately 31,5 metres  
 Your nearest friend(s) is/are at: C

Example of output if your friends are dining at restaurants C, F and H:

Restaurant C - approximately 22,5 metres  
 Restaurant H - approximately 22,5 metres  
 Restaurant F - approximately 31,5 metres  
 Your nearest friend(s) is/are at: C H

NOTE: You cannot move diagonally on the matrix.

MARK ALLOCATION FOR QUESTION 3.3

3	Locate the friends
4	Determine the name of the restaurant
6	Calculate the distance
3	Find the nearest friend(s)
2	Output to indicate position of friend(s) and distance
1	Correct format of output for the nearest friend(s)
<b>19</b>	<b>TOTAL</b>

- Enter your examination number as a comment in the first line of the program file.
- Save your program.
- Make a printout of the code if required.

TOTAL SECTION C: 37  
 GRAND TOTAL: 150



**Output area:**

Decide on a suitable output component to be placed in the output area provided to display the content of the two-dimensional array.

**NOTE:**

- You are NOT allowed to modify supplied data manually. Code must be written to manipulate the supplied data according to the requirements.
- Good programming techniques and modular design must be applied in the design and coding of your solution.
- **NO marks will be assigned for hardcoding.** Use control structures and variables where necessary.

**3.1 Button [Display layout]**

The program must display the layout of the mall. Write code to use the data supplied in the two-dimensional array to display the layout in neat columns.

Example of output:

A	X	O	*	O	X	E
B	X	O	O	O	X	D
C	X	O	O	O	X	C
D	X	O	O	O	X	D
E	X	O	O	O	X	E

(5)

Explanation of the data in the array:

- The characters A to J represent the position of each of the 10 restaurants in the mall.
- The character X is a placeholder indicating that none of your friends are currently in a restaurant.
- The character O indicates the open area where one can move around.
- The asterisk character (\*) indicates your position (first row and fourth column).

**NOTE:** You will always remain in the same position for the purpose of this question.

**3.2 Combo box with restaurant codes**

You are requested to select a restaurant from the combo box where one of your friends will dine.

Write code to replace the X placeholder at the selected restaurant with a hashtag symbol (#) to indicate that a friend is at the restaurant and display an updated layout of the mall.



**SECTION C**

**QUESTION 3: PROBLEM-SOLVING PROGRAMMING**

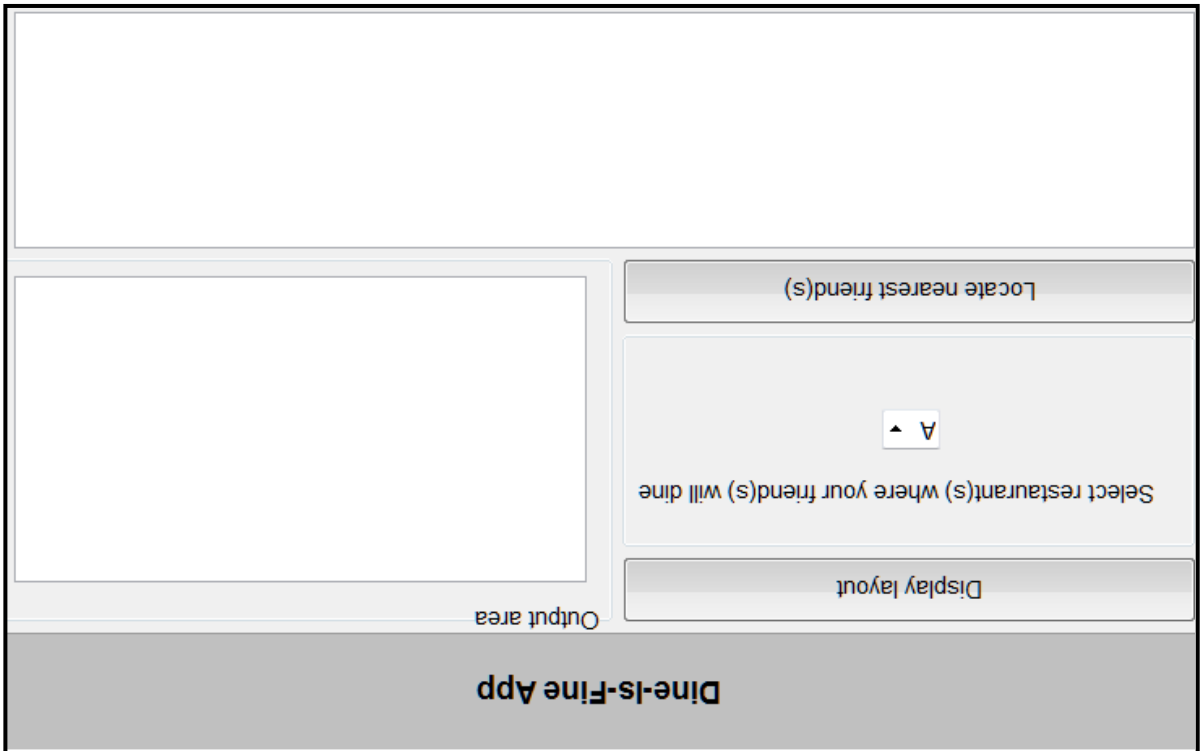
**SCENARIO**

A social media companion app called *Dine-Is-Fine* has been designed specifically for the 10 restaurants at your local mall. You and your friends are registered users of this app. The app can be used to share your location with other registered users.

Do the following:

- Compile and execute the program in the **Question3** folder. Currently the program has no functionality.
- Complete the code for each question, as described in QUESTION 3.1 to QUESTION 3.3.

**Supplied GUI:**



**Supplied data:**

You are provided with a two-dimensional array called **arrMall** filled with characters that represent the layout of the mall. The characters A to J represent 10 restaurants that you and your friends regularly visit.



Example of output for card number B870-4%-CH:

B870-4%-CH
Contact number: 0822001100
Updated number of loyalty points: 6519
Number of visits: 5
Health evaluation status: G

(3)

- Enter your examination number as a comment in the first line of the class and the form.
- Save all the files.
- Print the code contained in the class and the form (Delphi)/both classes (Java) if required.

**65 TOTAL SECTION B:**



- Use the method **isCorrect** to determine whether the access code that was entered is the correct access code for this card holder or not.

If the correct access code has been entered, use the data in the **DataJanuary2017.txt** text file to determine the following regarding the purchases of the card holder:

- The total amount spent for the month
- The amount spent on health-related products
- The number of times the user visited the shop and made purchases

Use this information and the relevant methods from the object class to update the number of loyalty points, the health level code and the number of times the card holder visited the shop for purchases. Enable the button for QUESTION 2.2.2.

If the access code is NOT correct, display a message to indicate that the login was not successful and clear the access code text box.

The data in the table below can be used for testing purposes.

Card number	Access code
B921-3\$-DI	188
B870-4%-CH	41
P730-7*-DI	338
B810-2#-DI	170

2.2.2

**Button – [2.2.2 – Display card holder details]**

Write code to use the **toString** method to display the card holder object data in the output area provided.

The method created in QUESTION 2.1.6 must be used to indicate if the card holder is a STAR shopper.

Example of output for card number B921-3\$-DI:

```

B921-3$-DI
Contact number: 0812345678
Updated number of loyalty points: 8190
Number of visits: 23
Health evaluation status: S
STAR shopper
    
```



2.2

Information on the visits and purchases of card holders is saved for one month only in a text file. The text file called **DataJanuary2017.txt** contains the transactions made by four of the card holders during January.

The contents of text file is as follows:

Every three lines represent a transaction for a specific loyalty card holder:

Line 1: The card number of the card holder, for example B921-3\$-DI  
Line 2: The total amount of the purchases for this visit, for example 1459.30  
Line 3: The amount spent on health food during this visit, for example 72.30  
Example of the first 12 lines of the text file **DataJanuary2017.txt**:

```
B921-3$-DI 1459.30 72.30
B870-4%-CH 899.80 22.50
F730-7*-DI 299.30 62.85
B870-4%-CH 299.30 62.85
F730-7*-DI 22.50 899.80
B870-4%-CH 72.30 1459.30
B921-3$-DI 182.55
```

Do the following to complete the code for each button in the main form unit (Delphi)/GUI class (Java), as in the section that follows.

**NOTE:** The object **objCardHolder** has been declared globally as part of the provided code.

2.2.1 Button – [2.2.1 – Check access code]

The user is required to select a card number from the four card numbers provided in the combo box.

Once a card number has been selected in the combo box, the cellphone number of the card holder and the number of loyalty points already earned from previous purchases are displayed with the provided code.

The user must enter the correct access code for the selected card number to allow the program to process the data of the card holder saved in the text file.

Write code to do the following:

- Use the card number, cellphone number and loyalty points of the card holder to instantiate a card holder object.



2.1.5 An access code is required to be able to update a customer's loyalty points. The customer's cellphone number has been used to compile the access code.

Write code to complete the **isCorrect** method provided. The method must test whether the access code received as a parameter is correct by applying the following instructions on the attribute for the cellphone number:

- Remove all zeros (0) from the cellphone number.
- Determine the sum of the remaining digits as follows:

○ If the number of remaining digits is even, every two consecutive digits starting on the left-hand side must be regarded as a two-digit value. Add all these two-digit values.

Example:

If the original cellphone number was 082 340 2113, the remaining digits without any zeros would be 82342113. The sum of the two-digit values is:

$$82 + 34 + 21 + 13 = 150$$

○ If the number of remaining digits is odd, the first digit must be regarded as a one-digit value, and thereafter every two consecutive digits must be regarded as a two-digit value. Add all these values.

Example:

If the original cellphone number was 082 034 2101, the remaining digits after removing the zeros are 8234211. The sum of the values is:

$$8 + 23 + 42 + 11 = 84$$

(13)

2.1.6 Write code for a method called **identifyStarShopper** that will return a string to indicate whether the card holder is a STAR shopper or not. A card holder is a STAR shopper if the following applies:

- The card holder earned more than 2 000 loyalty points and the shop was visited more than 10 times during the month.
- OR
- The health level of the card holder is platinum.

The string to be returned must either be 'STAR shopper' or an empty string.

(6)







**SECTION B**

**QUESTION 2: OBJECT-ORIENTATED PROGRAMMING**

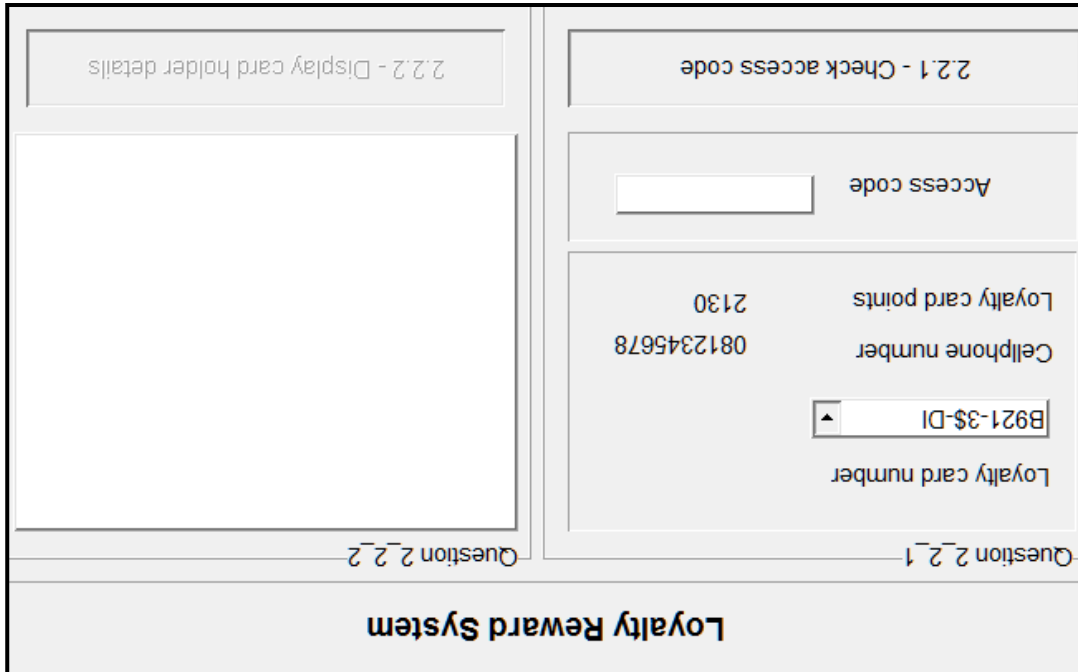
The local supermarket has a loyalty card system where customers can receive rewards depending on the number of visits, the number of loyalty points gathered and the percentage of health food bought.

Do the following:

Delphi programmers	Java programmers
<ul style="list-style-type: none"> <li>• Open the incomplete program in the <b>Question2</b> folder.</li> <li>• Open the incomplete object class <b>CardHolder_U.pas</b>.</li> <li>• Add your examination number as a comment in the first line of both <b>Question2_U.pas</b> and <b>CardHolder_U.pas</b>.</li> </ul>	<ul style="list-style-type: none"> <li>• Open the incomplete program in the <b>Question2</b> folder.</li> <li>• Open the incomplete object class <b>CardHolder.java</b>.</li> <li>• Add your examination number as a comment in the first line of both classes <b>Question2.java</b> and <b>CardHolder.java</b>.</li> </ul>

- Compile and execute the program. Currently the program has no functionality.

Example of graphical user interface (GUI):



- Complete the code for this program, as specified in QUESTION 2.1 and QUESTION 2.2 below.

2.1 The given incomplete object class (**TCardHolder/CardHolder**) contains the declaration of five attributes, the **toString()** method and an incomplete **isCorrect()** method.



1.4

**Button [Question 1\_4]**

The customer's ID number entered in QUESTION 1\_1 and the type of loyalty card selected in QUESTION 1\_2 are required to compile the customer's loyalty card number. A loyalty card number consists of three parts. Each part of the number must be compiled as follows and displayed in the components provided:

**Part 1:**

The first part of the loyalty card number consists of either the letter B if the business card check box was selected or the letter P if the business card check box was not selected, followed by the first three digits of the customer's ID number.

**Part 2:**

A string called **charstring**, which contains the following 10 characters, has been supplied as part of the code provided:

i @ # \$ % [ & \* ~ \ ^

Each character contained in the string represents a digit (0–9). This means that the first character (i) represents the digit 0, the second character (@) the digit 1, and so on.

This part of the loyalty card number must consist of the last digit from the customer's ID number, followed by the character from **charstring** that represents that digit.

Example of this part of the loyalty card number if the customer's ID number is 8802128756084:

4%

**Part 3:**

This part of the loyalty card number consists of the code representing the type of loyalty card (DI or CH) established in QUESTION 1\_2.

Example of the loyalty card number of a customer with the following profile:

- Application for a business
- ID number: 9210140012083
- Card type: DI

Your loyalty card number

B921 - 3\$ - DI

(13)

- Enter your examination number as a comment in the first line of the program file.
- Save your program.
- Make a printout of the code if required.

**TOTAL SECTION A: 48**



- A code for the type of loyalty card that was selected must be assigned to the global variable **typeCard** that is provided. The code DI represents immediate discount and the code CH represents a donation towards charity.

**NOTE:** Only the label for the currently selected button must appear in bold. If the user selects the other type of loyalty card, the label of the previously selected loyalty card must NOT appear in bold.

(6)

1.3

**Button [Question 1\_3]**

The final purchase amount for today must be entered in the text box.

The program must grant the following benefit towards the holder of the type of loyalty card selected in QUESTION 1\_2:

- Immediate discount (DI):  
1,5% discount must be granted on the purchase amount.
- Donation towards charity (CH):

An amount of R3 will be donated to a charity organisation for every R100 or part of R100 spent by the loyalty card holder.

Display a message describing the benefit granted depending on the type of loyalty card.

Example of output if the code of the loyalty card is DI and the purchase amount R1 200:

**NOTE:** The discount amount must be displayed as a currency with two decimal digits.

Example of output if the code of the loyalty card is CH and the purchase amount R230:

(15)



1.1

**Button [Question 1\_1]**

A label called `lblWelcomeMessage` has been created and set to be not visible. Write code to validate the information entered by the user as follows:

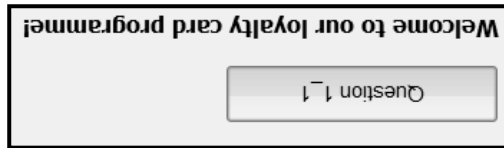
- Name and surname

The input must contain at least one space to be valid. If the input is not valid, display a message to indicate the specific error using a dialog box.

- ID number

The asterisk character (\*) next to the text box indicates that the field may not be left blank. An ID number must consist of 13 digits to be valid. If the ID number field does not contain any information or the ID number that was entered does not consist of 13 digits, display a message to indicate the specific error that occurred using a dialog box.

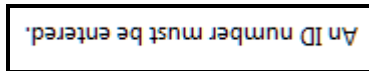
If the contents in both fields are valid, the label displaying the welcoming message must be set to visible.



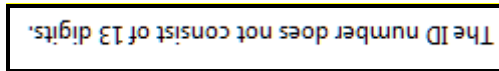
Example of message if no space was entered in the name-surname field:



Example of message if the ID number field was left blank:



Example of message if the ID number that was entered does not consist of 13 digits:



(14)

1.2

**Buttons with images [Question 1\_2]**

A customer must select the preferred type of loyalty card. The images on the buttons in this section of the GUI show two types of loyalty cards. When a specific image is selected (button clicked), the program must do the following:

- The heading in the label above the selected image must be changed to bold.



**SCENARIO**

Loyalty cards are often used to create profiles for customers. It is also a popular way of motivating customers to visit shops that regularly award loyalty points for purchases made.

**SECTION A**

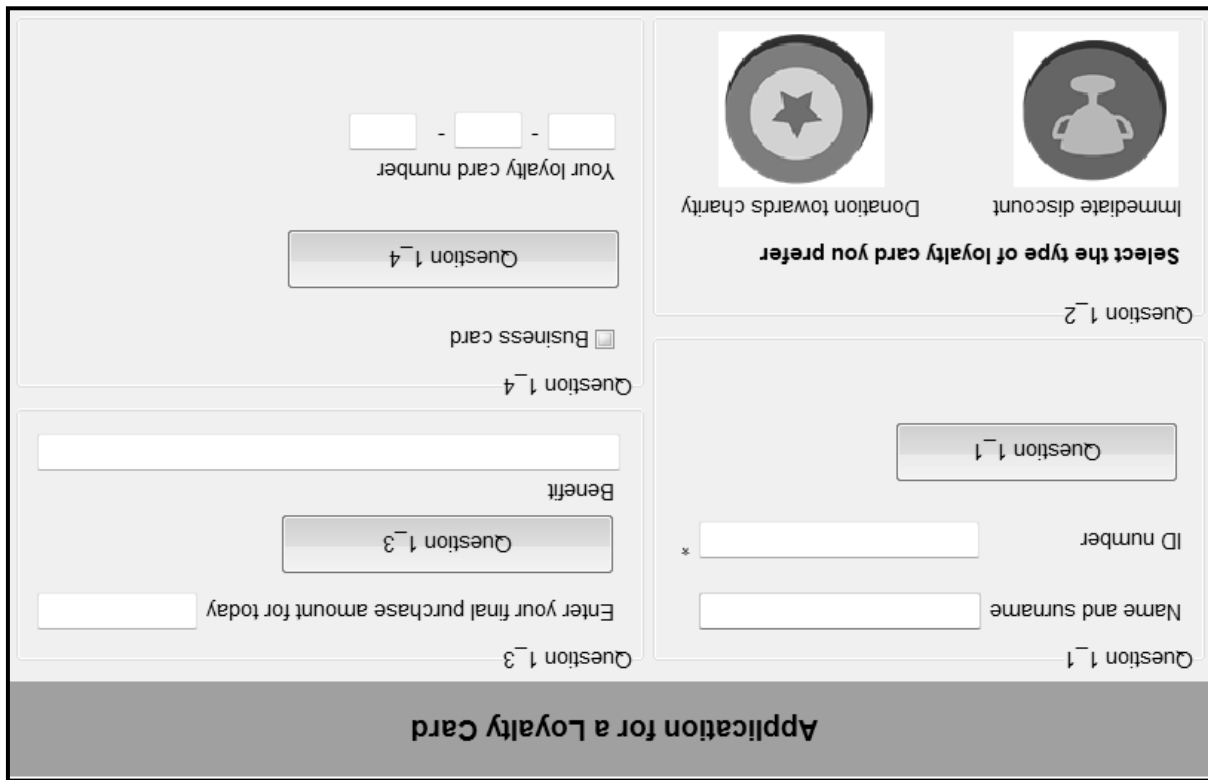
**QUESTION 1: GENERAL PROGRAMMING SKILLS**

An applicant must complete an application form to obtain a loyalty card.

Do the following:

- Compile and execute the program found in the **Question1** folder. The user interface displays four different sections named **QUESTION 1\_1** to **QUESTION 1\_4**. Currently the program has no functionality.

Example of graphical user interface (GUI):



- Complete the code for each section of **QUESTION 1**, as described in **QUESTION 1.1** to **QUESTION 1.4** on the next page.



13.

The files that you need to complete this question paper have been given to you on the disk/CD/DVD/flash disk or on the disk space allocated to you. The files are provided in the form of password-protected executable files.

**NOTE:**

- Delphi candidates must use the file **DelphiDataENGMarch2017.exe**.
- Java candidates must use the file **JavaDataENGMarch2017.exe**.

Do the following:

- Double click on the password-protected executable file.
- Click on the extract button.
- Enter the following password: **Prof4U@#!**

Once extracted, the following list of files will be available in the folder **DelphiDataENGMarch2017/JavaDataENGMarch2017:**

**JAVA (NETBEANS) FILES**

**Question1:**  
Question1.form  
Question1.java  
Type1.PNG  
Type2.PNG

**Question1:**  
Question1\_P.dpr  
Question1\_P.dproj  
Question1\_P.res  
Question1\_U.dfm  
Question1\_U.pas  
Type1.PNG  
Type2.PNG

**Question2:**  
CardHolder\_U.pas  
DataJanuary2017.txt  
Question2\_P.dpr  
Question2\_P.dproj  
Question2\_P.res  
Question2\_U.dfm  
Question2\_U.pas

**Question2:**  
CardHolder.java  
DataJanuary2017.txt  
Question2.form  
Question2.java

**Question3:**  
Question3\_P.dpr  
Question3\_P.dproj  
Question3\_P.res  
Question3\_U.dfm  
Question3\_U.pas

**Question3:**  
Question3.form  
Question3.java



**INSTRUCTIONS AND INFORMATION**

1. This question paper is divided into THREE sections. Candidates must answer ALL THREE sections.
2. The duration of this examination is three hours. Because of the nature of this examination it is important to note that you will not be permitted to leave the examination room before the end of the examination session.
3. This question paper is set with programming terms that are not specific to any particular programming language (Delphi/java (using the Netbeans IDE)).
4. Make sure that you answer the questions according to the specifications that are given in each question. Marks will be awarded according to the set requirements.
5. Answer only what is asked in each question. For example, if the question does not ask for data validation, then no marks will be awarded for data validation.
6. Your programs must be coded in such a way that they will work with any data and not just the sample data supplied or any data extracts that appear in the question paper.
7. Routines, such as search, sort and selection, must be developed from first principles. You may NOT use the built-in features of a programming language for any of these routines.
8. All data structures must be defined by you, the programmer, unless the data structures are supplied.
9. You must save your work regularly on the disk/CD/DVD/flash disk you have been given, or on the disk space allocated to you for this examination session.
10. Make sure that your examination number appears as a comment in every program that you code, as well as on every event indicated.
11. If required, print the programming code of all the programs/classes that you completed. You will be given half an hour printing time after the examination session.
12. At the end of this examination session you must hand in a disk/CD/DVD/flash disk with all your work saved on it OR you must make sure that all your work has been saved on the disk space allocated to you for this examination session. Ensure that all files can be read.





# MORNING SESSION

This question paper consists of 16 pages.

TIME: 3 hours

MARKS: 150

FEBRUARY/MARCH 2017

INFORMATION TECHNOLOGY P1

INF1.1

GRADE 12

NATIONAL  
SENIOR CERTIFICATE



Department:  
Basic Education  
REPUBLIC OF SOUTH AFRICA

basic education