



# basic education

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

## NATIONAL SENIOR CERTIFICATE

**GRADE 12**

**INFORMATION TECHNOLOGY P1**

**FEBRUARY/MARCH 2015**

**MARKS: 150**

**TIME: 3 hours**

**This question paper consists of 19 pages.**



\* I N F D M 1 \*



**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 paper is set in 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 only 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. You may not use components provided within the interface to store and later retrieve data.
9. You must save your work regularly on the disk you have been given, or the disk space allocated to you for this examination.
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.



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

- Delphi learners must use the file **DelphiDataENG.exe**.
- Java learners must use the file **JavaDataENG.exe**.

Do the following:

- Double click on the file.
- Click on the extract button.
- Enter the following password: **ProperTies87&%**

Once the file has been extracted, the following list of files will be available in the folder **DelphiDataENG/JavaDataENG**:

#### **Delphi files**

##### **Question1:**

Question1\_P.dpr  
Question1\_P.res  
Question1\_U.dfm  
Question1\_U.pas

##### **Question2:**

Abdul Brouwer.txt  
HomeLoan\_U.pas  
Paula Redhat.txt  
Question2\_P.dpr  
Question2\_P.res  
Question2\_U.dfm  
Question2\_U.pas  
Xavier Outland.txt

##### **Question3:**

Question3\_P.dpr  
Question3\_P.res  
Question3\_U.dfm  
Question3\_U.pas

#### **Java (Netbeans) files**

##### **Question1:**

metre1.png  
Question1.form  
Question1.java

##### **Question2:**

Abdul Brouwer.txt  
Paula Redhat.txt  
Xavier Outland.txt

HomeLoan.java  
Question2.form  
Question2.java

##### **Question3:**

PopulateArrays.java  
Question3.form  
Question3.java

**SCENARIO**

You are employed as a software developer at a company that buys, renovates and sells properties.

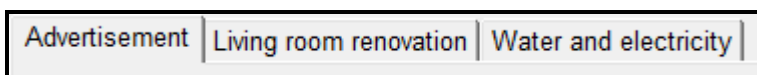
**SECTION A****QUESTION 1: GENERAL PROGRAMMING SKILLS****INSTRUCTIONS:**

<b>Delphi programmers</b>	<b>Java programmers</b>
<ul style="list-style-type: none"> <li>The project <b>Question1</b> is provided in the <b>Delphi</b> folder.</li> <li>Open the incomplete project file <b>Question1_P.dpr</b> in the <b>Question1</b> folder.</li> <li>Add your examination number as a comment in the first line of the main form unit <b>Question1_U.pas</b>.</li> </ul>	<ul style="list-style-type: none"> <li>The project <b>Question1</b> is provided in the <b>Netbeans</b> folder.</li> <li>Open the incomplete class called <b>Question1.java</b> in the folders <b>Source Packages (src)</b> and <b>Question1Package</b>.</li> <li>Add your examination number as a comment in the first line of the class <b>Question1.java</b>.</li> </ul>

Do the following:

- Compile and execute the program. The interface displays three different tabs, namely Advertisement, Living room renovation and Water and electricity.

Example of the tab sheets:



- Complete the code for each tab as described in QUESTION 1.1 to QUESTION 1.3.

**NOTE:** QUESTION 1.1 refers to the tab Advertisement.  
 QUESTION 1.2 refers to the tab Living room renovation.  
 QUESTION 1.3 refers to the tab Water and electricity.

### 1.1 Interface for the Advertisement tab

Market value: 0

Selling price: 0

Number of bedrooms: 0

Number of bathrooms: 0

☐ Pool

Generate advertisement

Advertisement

**NOTE:** The **market value** of a property is the municipal value that the property is assessed for.

The **selling price** of a property is the value of the sale of the property.

#### [Button – Generate advertisement]

Write code to do the following:

- Use information from the provided text boxes to compile an advertisement in the following format:

```
House for sale:  
<selling price>#<number of bedrooms>Bed#<number of bathrooms>Bath#
```

Use the following information to compile the advertisement:

- The selling price contained in the advertisement must be formatted to a currency. Do not display any decimal values.
- If the Pool check box is selected, the text 'Pool#' must be added to the advertisement text.
- If the selling price of the property is less than the market value, the word 'Bargain' must be added to the advertisement text.

- Display the advertisement information in the output area provided.

Examples of compiled advertisements:

Example 1: If the market value of the property is R950 000, the selling price is R850 000 and the house has three bedrooms, two bathrooms and a pool, the advertisement should be displayed as follows:

House for sale:  
R850000#3Bed#2Bath#Pool#Bargain

Example 2: If the market value of the property is R750 000, the selling price is R800 000 and the house has two bedrooms, one bathroom and no pool, the advertisement should be displayed as follows:

House for sale:  
R800000#2Bed#1Bath#

(11)

## 1.2 Interface for the Living room renovation tab

Write code to do the following:

### 1.2.1 [Radio Button – Painting]

If the radio button Painting is selected, use a variable to save the selected type of renovation as the character P.

(1)

**1.2.2 [Radio Button – Tiling]**

If the radio button Tiling is selected, use a variable to save the selected type of renovation option as the character T.

(1)

**1.2.3 [Button – Calculate and display renovation cost]**

The user is required to enter the number of square metres to be renovated as an integer value and select the type of renovation to be done. The cost of each type of renovation must be calculated as explained below.

(a) If the type of renovation is **Painting**, use the following information to calculate how much paint will be required as well as the total cost:

- One litre of paint is used to paint an area of eight square metres.
- The pricing structure of paint per drum is as follows:
  - 1-litre drum R 55.50
  - 2-litre drum R 92.30
  - 5-litre drum R 199.00

Display the calculated values in the provided output area using the following format:

Area: `<area>` square metres

Volume of paint required: `<volume>` litre

1-litre drums: `<number of 1-litre drums>`

2-litre drums: `<number of 2-litre drums>`

5-litre drums: `<number of 5-litre drums>`

Total cost: `<Total cost of paint>`

The total cost must be displayed rounded to two decimal places.

Example: If the area entered is 148 m<sup>2</sup>, the output will be as follows:

Area: 148 square metres

Volume of paint required: 18.5 litres

1-litre drums: 0

2-litre drums: 2

5-litre drums: 3

Total cost: R 781.60

(17)



- (b) If the type of renovation is **Tiling**, a dialog box must be used to allow the user to enter the cost of tiling per square metre. Add five square metres to the original square metre value that was entered to provide for breakages, and then calculate the cost. Display the area to be tiled in square metres. The total cost must be displayed rounded to two decimal places.

Example: If the area entered is 100 m<sup>2</sup> and the cost of tiling is R50,00 per square metre, the following will be displayed:

Area: 100 square metres  
Total cost: R 5250.00

(6)

### 1.3 Interface for the Water and electricity tab

The screenshot shows a software interface with two main sections. The top section, titled 'Calculate electricity used', contains a label 'Electricity usage in kilowatts:' followed by two input fields: 'Previous reading' (containing '2100') and 'Current reading' (empty). To the right of these fields is a button labeled 'Calculate amount due' and a label 'Amount due' with the value '0.00'. The bottom section, titled 'Solar geyser options', contains a label 'Enter geyser size (50, 100, 150)' followed by an empty input field. Below this is a button labeled 'List geysers' and a large empty rectangular box.

Write code to do the following:

#### 1.3.1 [Button – Calculate amount due]

The previous reading for electricity used is supplied in a text box as an integer. The user must enter the current reading in a text box as an integer.

- If the current reading entered by the user is less than the previous reading, display a suitable message and clear the current reading text box.



- If the current reading is equal to or greater than the previous reading, use the following tariffs to calculate the amount due for electricity used:

Units	Tariffs
0–600	R1,00 per unit
>600	R600 + R1,50 per unit for all units that exceed 600

- Display the amount in the label provided in a format that includes a currency correct two decimal places.

Example:

Previous reading	Current reading		Amount due
<input type="text" value="2100"/>	<input type="text" value="2735"/>	<input type="button" value="Calculate amount due"/>	R 652.50

(11)

### 1.3.2 [Button – List geysers]

In an effort to save electricity, solar geysers are investigated as a possibility. Three sizes of solar geysers are available based on their capacity in litres. The user is required to enter one of the three sizes: 50 (litres), 100 (litres) or 150 (litres).

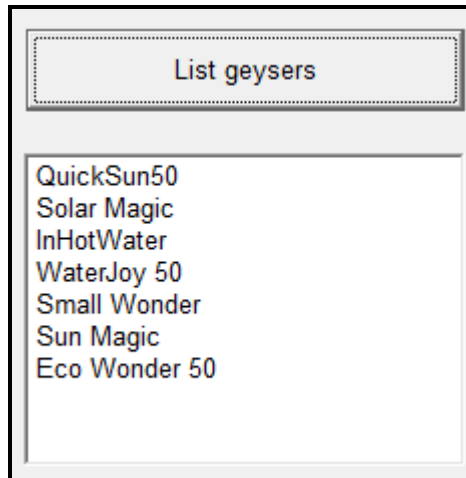
An array that contains strings of text which each describes a type of geyser is provided as part of the supplied code.

The description of each type of geyser starts with the capacity of the geyser in litres, followed by a dash and the brand name of the geyser.

Example: The geyser described as '50-Small Wonder' can hold 50 litres of water and the brand name of the geyser is 'Small Wonder'.

Write code to display the types of geysers in the array that matches the capacity entered by the user when the **List geysers** button is clicked. No validation of the input value is required.

Example: List of geysers when the value 50 is entered:



List geysers

- QuickSun50
- Solar Magic
- InHotWater
- WaterJoy 50
- Small Wonder
- Sun Magic
- Eco Wonder 50

(6)

- Ensure that your examination number is entered as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

**TOTAL SECTION A: 53**

**SECTION B****QUESTION 2: OBJECT-ORIENTATED PROGRAMMING**

Clients who want to buy properties usually apply for home loans. The following information about submitted applications are usually required:

- Does the applicant qualify for a home loan?
- What is the status of the application based on the duration of the loan in years and the interest rate?
- What is the instalment amount to be paid each month?

**INSTRUCTIONS:**

<b>Delphi programmers</b>	<b>Java programmers</b>
<ul style="list-style-type: none"> <li>• The project <b>Question2</b> is provided in the <b>Delphi</b> folder which contains: <ul style="list-style-type: none"> <li>○ A main form file called <b>Question2_U.pas</b></li> <li>○ An incomplete unit file called <b>HomeLoan_U.pas</b></li> <li>○ A text file for each applicant</li> </ul> </li> <li>• Open the incomplete project file called <b>Question2_P.dpr</b> in the <b>Question2</b> folder.</li> <li>• View (Ctrl+F12) the unit file <b>HomeLoan_U.pas</b> and add your examination number as a comment in the first line of both unit files <b>Question2_U.pas</b> and <b>HomeLoan_U.pas</b>.</li> </ul>	<ul style="list-style-type: none"> <li>• The project <b>Question2</b> is provided in the <b>Netbeans</b> folder which contains: <ul style="list-style-type: none"> <li>○ A GUI class file called <b>Question2.java</b></li> <li>○ An incomplete object class called <b>HomeLoan.java</b></li> <li>○ A text file for each applicant</li> </ul> </li> <li>• Open the incomplete file called <b>Question2.java</b> and <b>HomeLoan.java</b> in the folders <b>Source Packages (src)</b> and <b>Question2Package</b>.</li> <li>• Add your examination number as a comment in the first line of the classes <b>Question2.java</b> and <b>HomeLoan.java</b>.</li> </ul>

Do the following:

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

Example of the interface (see next page):

Home loan applications

Names of possible applicants

Find applicant

Evaluate application status

Years:

Interest rate (%):

Evaluate application

- Complete the code as described in QUESTION 2.1 and QUESTION 2.2 to add functionality to the program.

2.1 Complete the code in the home loan class **THomeLoan/HomeLoan** as described below.

2.1.1 Write code to add the following five attributes to the class:

Description	Names of attributes	
	Delphi	Java
Name of applicant	fApplicantName	applicantName
Disposable income	fDisposableIncome	disposableIncome
Home loan amount applied for	fLoanAmount	loanAmount
Number of years to repay the home loan	fYears	years
Interest rate	fInterestRate	interestRate

(4)

2.1.2 Write code to create a constructor which receives the following parameter values:

name of applicant  
disposable income  
loan amount applied for

Initialise the relevant attributes using the parameter values. Initialise the attributes for the number of years and the interest rate to 0.

(5)

- 2.1.3 Write an accessor method for the name of the applicant. (2)
- 2.1.4 Write mutator methods for the attributes Years and Interest rate. (4)
- 2.1.5 A method called **calculateInstalment** has been provided. Uncomment the code that calculates the monthly instalments in order to make the method functional. (1)
- 2.1.6 Write code for a method called **isApproved** to return a Boolean value indicating whether the home loan has been approved or not.
- A home loan will only be approved if the following criteria are met:
- Loans of more than R800 000 must be repaid over a period of more than 20 years (25 or 30 years).
  - The monthly disposable income of an applicant who applies for a loan up to R600 000 must be at least 130% of the monthly instalments.
- 2.1.7 Write code to create a **toString** method to return a string formatted as follows: (9)

Name of applicant: <applicant name>  
 Disposable income: <disposable income>  
 Loan amount: <loan amount>  
 Number of years: <number of years>  
 Interest rate: <interest rate> %

(5)

- 2.2 You have been provided with **three text files** which each contains information about a different home loan application. The name of each text file is the name of the applicant.

Example: If the applicant's name is 'Abdul Brouwer', the name of the text file is **Abdul Brouwer.txt**.

Each text file contains four lines of information about a home loan applicant in the following format:

<name and surname of applicant>  
 <monthly income>  
 <monthly expenditure>  
 <home loan amount applied for>

Content of the three text files for each of the following applicants:  
 Abdul Brouwer, Paula Redhat and Xavier Outland

Abdul Brouwer	Paula Redhat	Xavier Outland
28000	25785	44741
12500	11190	20384
750000	888000	978000



The GUI is supplied with the names of different applicants stored in the combo box that is provided. An object of the **THomeLoan/HomeLoan** class has been declared globally.

### 2.2.1 [Button – Find applicant]

The user is required to select the name of an applicant from the combo box.

Write code to do the following:

- Test whether an applicant was selected or not and display a suitable message.
- Test whether a text file with the applicant's name exists or not, and display a suitable message which includes the applicant's name, and whether an application was submitted or not.
  - If a text file is not found for the applicant selected from the combo box, no home loan application was submitted.

Example of output of a home loan application by John Smith that was not submitted:

No home loan application was submitted by John Smith.

If a home loan application was not submitted, disable the 'Evaluate application' button.

- If a text file with the applicant's name exists (was submitted), do the following:
  - Read the income, expenditure and loan amount from the text file.
  - Calculate the disposable income by subtracting the expenses from the income.
  - Instantiate a **THomeLoan/HomeLoan** object for the applicant.
  - Enable the button 'Evaluate application'.

(18)

### 2.2.2 [Button – Evaluate application]

The user must select the number of years for the loan to be repaid from the combo box provided and enter the interest rate in the text box provided for this purpose.

Write code to do the following:

- Use the mutator methods and the information provided by the user to update the applicable attributes of the home loan object for the applicant.

- Test whether the loan was approved (adhered to the requirements) or not.
  - If the loan was approved, display the details of the loan application in the output area using the **toString** method. Display a message with the following format:

Loan APPROVED with a monthly instalment of `<monthly instalment>`

Example of output for the approved home loan application by Abdul Brouwer for R750 000 over 20 years at an interest rate of 9%:

```
Name of applicant: Abdul Brouwer
Disposable income: R 15500.00
Loan amount: R 750000.00
Number of years: 20
Interest rate: 9.0%

Loan APPROVED with a monthly instalment of
R 6847.00
```

- If the home loan was not approved, display the applicant's name and surname and a message indicating that the loan was not approved.

Example of output for the home loan application by Xavier Outland for R870 000 over 20 years at an interest rate of 9%, which was not approved:

```
Name of applicant: Xavier Outland
Loan NOT APPROVED
```

(12)

- Ensure that your examination number is entered as a comment in the first line of the class and the form.
- Save all the files.
- A printout of the code in both classes/units may be required.

**TOTAL SECTION B: 60**

**SECTION C****QUESTION 3: PROBLEM-SOLVING PROGRAMMING**

Six estate agents are employed at a local estate agency. The manager of the agency requires software that is able to evaluate the profiles of their sales agents annually.

**INSTRUCTIONS:**

Delphi programmers	Java programmers
<ul style="list-style-type: none"> <li>The project <b>Question3</b> is provided in the <b>Delphi</b> folder.</li> <li>Open the incomplete project file <b>Question3_P.dpr</b> in the <b>Question3</b> folder.</li> <li>Add your examination number as a comment in the first line of the main form unit file <b>Question3_U.pas</b>.</li> </ul>	<ul style="list-style-type: none"> <li>The project <b>Question3</b> is provided in the <b>Netbeans</b> folder.</li> <li>Open the incomplete classes called <b>Question3.java</b> and <b>PopulateArrays.java</b> in the folders <b>Source Packages (src)</b> and <b>Question3Package</b>.</li> <li>Add your examination number as a comment in the first line of the class file <b>Question3.java</b>.</li> </ul>

Do the following:

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

Example of interface:

Agent's sales details

Enter agent's code

Find agent's name

Agent's sales

Display area





Two arrays called **arrAgents** and **arrSales** are supplied. You may create and use additional arrays as part of your solution.

The array **arrAgents** stores the codes and names of a number of sales agents in the following format:

**<Agent's code>:<Agent's name>**

Example:

A120:Wes Seelig

**NOTE:** The information of sales agents with agent's codes A120–A125 is supplied.

The array **arrSales** stores information about sold properties in the following format:

**<month>#<type>#<agent's code>;<price>**

Description of information provided in one line of text:

- **month:** The number of the month of the year in which the sale transaction was conducted
- **type:** The type of property sold:
  - **C:** Commercial
  - **R:** Residential
  - **A:** Agricultural
- **agent's code:** The code of the agent who sold the property
- **price:** The price at which the property was sold

Example:

02#C#A123;650000

- Complete the code for QUESTION 3 as described in QUESTION 3.1 and QUESTION 3.2 below.

### 3.1 [Button – Find agent's name]

The user is required to enter the code for an agent in the text box provided.

Write code that will search the **arrAgents** array for the code of the agent that was entered by the user. The search process should be constructed in such a way that the search process stops when the agent's code is found.

If the agent's code is found, display the agent's name in the label provided for the name of the agent.

If the agent's code is not found, display the message 'Invalid agent's code' in the label which is provided for the name of the agent and clear the text box provided for entering the agent's code.

Example of output if the agent's code entered is A123:

Agent's sales details

Enter agent's code

A123

Find agent's name

Melda Bissonette

Agent's sales

Example of output if the agent's code entered is B123:

Agent's sales details

Enter agent's code

Find agent's name

Invalid agent code

Agent's sales

(11)

### 3.2 [Button – Agent's sales]

3.2.1 Add a suitable output component in the **Display area** on the GUI to provide for the output for QUESTION 3.2.2. Rename the component with an appropriate name. (2)

3.2.2 A summary of the sales conducted by the sales agent identified in QUESTION 3.1 must be displayed in the component added in the GUI in QUESTION 3.2.1. Use the information in the **arrSales** array and summarise the sales of the agent as shown below.

Example of output if the agent's code is A123:

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
Commercial	1	3	3	5	0	0	1	2	0	2	1	6	24
Residential	0	0	1	0	0	1	0	0	0	0	0	0	2
Agricultural	0	0	3	4	0	0	3	1	3	0	0	0	14
Total value of sales: R 79640000													

**NOTE:** The values displayed in the last column indicate the total number of the different types of properties sold by this agent.

The total value of sales displayed below the summary indicates the total value of all properties sold by this agent.

Any other suitable output component may be used other than the one used in the given example.

(24)

- Ensure that your examination number is entered as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

**TOTAL SECTION C: 37**  
**GRAND TOTAL: 150**



Die totale waarde van die verkope wat onderkant die opsomming vertoon word, dui die totale waarde aan van al die eiendomme wat deur hierdie agent verkoop is.

Enige ander geskikte afvoerkomponent kan gebruik word behalwe die een wat in die gegewe voorbeeld gebruik is.

(24)

- Maak seker dat jou eksamennummer as kommentaar in die eerste reël van die programmeer ingesleutel is.
- Stoor jou program.
- 'n Drukstuk van die kode mag vereis word.

**TOTAAL AFDELING C: 37**  
**GROOTTOTAAL: 150**



Voorbeeld van afvoer as die agentkode wat ingesleutel is, A123 is:

Agent se besonderhede van verkope

Sleutel agent se kode in

Melda Bissonette

Agent se verkope

Vind agent se naam

Voorbeeld van afvoer as die agentkode wat ingesleutel is, B123 is:

Agent se besonderhede van verkope

Sleutel agent se kode in

Ongeldige agentkode

Agent se verkope

Vind agent se naam

3.2

[Knoppie – Agent se verkope]

(2)

3.2.1

Voeg 'n geskikte afvoerkomponent by in die **Vertoon-area** op die GSK ('GUI') om voorsiening te maak vir die afvoer vir VRAAG 3.2.2. Herneem die komponent met 'n geskikte naam.

3.2.2

'n Opsomming van die verkope wat gedoen is deur die agent wat in VRAAG 3.1 geïdentifiseer is, moet in die komponent wat in die GSK ('GUI') van VRAAG 3.2.1 bygevoeg is, vertoon word. Gebruik die inligting in die **arrVerkope**-skikking en som die verkope van die agent op soos hieronder aangetoon.

Voorbeeld van afvoer as die agent se kode A123 is:

	Jan	Feb	Mrt	Apr	Mei	Jun	Jul	Aug	Sep	Okt	Nov	Des
Kommersieel	1	3	3	5	0	0	1	2	0	2	1	6
Residensieel	0	0	1	0	0	1	0	0	0	0	0	2
Landbou	0	0	3	4	0	0	3	1	3	0	0	14
Totale waarde van verkope: R 79640000												

**LET WEL:** Die waardes wat in die laaste kolom vertoon word, dui die totale getal van die verskillende tipes eiendomme wat deur hierdie agent verkoop is, aan



Twee skikkinge met die name **arrAgente** en **arrVerkope** is voorsien. Jy mag addisionele skikkinge as deel van jou oplossing skep en gebruik.

Die skikking **arrAgente** stoor die kodes en name van 'n aantal verkoopsagente in die volgende formaat:

```
<Agent se kode>:<Agent se naam>
```

Voorbeeld:

A120:Wes Seelig

**LET WEL:** Die inligting van verkoopsagente met agentkodes A120-A125 is voorsien.

Die skikking **arrVerkope** stoor inligting oor eiendomme wat verkoop is in die volgende formaat:

```
<maand>#<tipe>#<agent se kode>:<prys>
```

Beskrywing van inligting wat in een teksreël voorsien word:

- **maand:** Die nommer van die maand van die jaar waarin die verkooptransaksie geding is
- **tipe:** Die tipe eiendom wat verkoop is:

<b>K:</b>	Kommersieel
<b>R:</b>	Residensieel
<b>L:</b>	Landbou

- **agent se kode:** Die kode van die agent wat die eiendom verkoop het
- **prys:** Die prys waarteen die eiendom verkoop is

Voorbeeld:

02#K#A123;650000

- Voltooi die kode vir VRAAG 3 soos wat dit hieronder in VRAAG 3.1 en VRAAG 3.2 beskryf is.

### 3.1 [Knoppie – Vind agent se naam]

Die gebruiker moet die kode van 'n agent in die teksblokkie ('text box') wat voorsien is, insleutel.

Skryf kode wat die agent van die kode wat deur die gebruiker ingesleutel is, in die **arrAgente**-skikking sal soek. Die soekproses moet op so 'n manier saamgestel word dat die soekproses sal staak wanneer die agent se kode gevind word.

As die agent se kode gevind word, vertoon die agent se naam in die byskrif ('label') wat vir die naam van die agent voorsien is.

As die agent se kode nie gevind word nie, vertoon die boodskap 'Ongeldige agentkode' in die byskrif ('label') wat vir die naam van die agent voorsien is en maak die teksblokkie ('text box') wat vir die insleutel van die agentkode voorsien is, skoon.



AFDELING C

VRAAG 3: PROBLEEMOPLOSSINGPROGRAMMERING

Ses eiendomsagente is deur 'n plaaslike eiendomsagentskap in diens geneem. Die bestuurder van die agentskap wil sagteware hê wat in staat is om die profiele van hulle verkoopsagente jaarliks te evalueer.

INSTRUKSIES:

<b>Delphi-programmeerders</b>	<ul style="list-style-type: none"><li>• Die projek <b>Vraag3</b> is in die <b>Delphi-</b> lêrigds voorsien.</li><li>• Maak die onvlottoide projekleër <b>Vraag3_P.dpr</b> in die <b>Vraag3-lêrigds</b> oop.</li><li>• Voeg jou eksamennummer as kommentaar in die eerste reël van die hoofvormeenheid-lêër <b>Vraag3_U.pas</b> by.</li></ul>
<b>Java-programmeerders</b>	<ul style="list-style-type: none"><li>• Die projek <b>Vraag3</b> is in die <b>Netbeans-</b> lêrigds voorsien.</li><li>• Maak die onvlottoide klasse met die name <b>Vraag3.java</b> en <b>Vuisikikings.java</b> in die lêrigds <b>Source Packages (src)</b> en <b>Vraag3Package</b> oop.</li><li>• Voeg jou eksamennummer as kommentaar in die eerste reël van die klasleër <b>Vraag3.java</b> by.</li></ul>

Doen die volgende:

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

Voorbeeld van koppelvlak:

Agent se besonderhede van verkope

Sleutel agent se kode in

Find agent se naam

Agent se verkope

Vertoon-area



- Toets of die lening goedgekeur is (aan die vereistes voldoen het) of nie.

○ As die lening goedgekeur is, vertoon die besonderhede van die leningaansoek in die afvoerarea deur die **tostring-**metode te gebruik. Vertoon 'n boodskap met die volgende formaat:

Lening GOEDGEKEUR met 'n maandelikse paaiement van <maandelikse paaiement>

Voorbeeld van afvoer vir die goedgekeurde huisleningaansoek deur Abdul Brouwer vir R750 000 oor 20 jaar teen 'n rentekoers van 9%:

Naam van aansoeker: Abdul Brouwer Besteebare inkomste: R 15500.00 Leningbedrag: R 750000.00 Getal jaar: 20 Rentekoers: 9.0% Lening GOEDGEKEUR met 'n maandelikse paaiement van R 6847.00
---

○ As die huislening nie goedgekeur is nie, vertoon die aansoeker se naam en van en 'n boodskap wat aandui dat die lening nie goedgekeur is nie.

Voorbeeld van afvoer vir 'n huisleningaansoek deur Xavier Outland vir R870 000 oor 20 jaar teen 'n rentekoers van 9% wat nie goedgekeur is nie:

Naam van aansoeker: Xavier Outland Lening NIE GOEDGEKEUR NIE
---

(12)

- Maak seker dat jou eksamennummer as kommentaar in die eerste reël van die klas en die vorm verskyn.
- Stoor al die lêers.
- 'n Drukstuk van die kode vir beide klasse/eenhede mag vereis word.

## TOTAAL AFDELING B: 60





Die GJK ('GUI') is voorsien van die name van die verskillende aansoekers wat gestoor is in die kombinasieys ('combo box') wat voorsien is. 'n Objek van die **ThuisLening/HuisLening**-klas is globaal verklaar.

## 2.2.1 [Knoppie – Vind aansoeker]

Die gebruiker moet die naam van 'n aansoeker uit die kombinasieys ('combo box') kies.

Skryf kode om die volgende te doen:

- Toets of 'n aansoeker gekies is of nie en vertoon 'n geskikte boodskap.
- Toets of 'n teksleër met die aansoeker se naam bestaan of nie, en vertoon 'n geskikte boodskap wat die naam van die aansoeker insluit, en of 'n aansoek ingedien is of nie.

- As 'n teksleër nie gevind word vir die aansoeker wat in die kombinasieys ('combo box') gekies is nie, is geen aansoek vir 'n huislening ingedien nie.

Voorbeeld van afvoer van 'n huislening-aansoek deur John Smith wat nie ingedien is nie:

**Geen huislening-aansoek is deur John Smith ingedien nie.**

As 'n huislening-aansoek nie ingedien is nie, maak die 'Evalueer aansoek'-knoppie ('button') onaktief.

- As 'n teksleër met die aansoeker se naam bestaan (ingedien is), doen die volgende:

- Lees die inkomste, uitgawes en leningbedrag uit die teksleër.
- Bereken die besteebare inkomste deur die uitgawes van die inkomste af te trek.
- Instansieer 'n **ThuisLening/HuisLening**-objek vir die aansoeker.
- Maak die 'Evalueer aansoek'-knoppie ('button') aktief.

(18)

## 2.2.2 [Knoppie – Evalueer aansoek]

Die gebruiker moet die getal jaar vir die lening om terugbetaal te word, kies uit die kombinasieys ('combo box') wat voorsien is en die rentekoers in die teksblokkie wat vir hierdie doel voorsien is, insleutel.

Skryf kode om die volgende te doen:

- Gebruik die wysigingsmetodes en die inligting wat deur die gebruiker voorsien is om die toepaslike attribute van die huisleningsobjek van die aansoeker op te dateer.



- 2.1.3 Skryf 'n toegangsmetode ('accessor method') vir die naam van die aansoeker. (2)
- 2.1.4 Skryf wysigingsmetodes ('mutator methods') vir die attribute Jare en Rentekoers. (4)
- 2.1.5 'n Metode met die naam **BerekenPaaiement** is voorsien. Dekommenteer ('Uncomment') die kode wat die maandelikse paaiement bereken sodat die metode funksioneel gemaak word. (1)
- 2.1.6 Skryf kode vir 'n metode met die naam **isGoedgekeur** wat 'n Boolese waarde, wat aandui of die huislening goedgekeur is of nie, sal terugstuur. 'n Huislening sal slegs goedgekeur word as daar aan die volgende kriteria voldoen word:
- Lenings van meer as R800 000 moet oor 'n tydperk van meer as 20 jaar (25 of 30 jaar) terugbetaal word.
  - Die maandelikse besteebare inkomste van 'n aansoeker wat vir 'n lening van tot R600 000 aansoek doen, moet ten minste 130% van die maandelikse paaiement wees.
- Skryf kode om 'n **tostring**-metode te skep om 'n string terug te stuur wat soos volg geformateer is:
- Naam van aansoeker: <aansoeker se naam>  
 Besteebare inkomste: <besteebare inkomste>  
 Leningbedrag: <Leningbedrag>  
 Getal jaar: <getal jaar>  
 Rentekoers: <rentekoers> %
- (5)

2.2

Jy is voorsien van **drie teksleërs** wat elk inligting oor 'n ander huisleningaansoek bevat. Die naam van elke teksleër is die naam van die aansoeker.

Voorbeeld: As die aansoeker se naam 'Abdul Brouwer' is, is die naam van die teksleër **Abdul Brouwer.txt**.

Elke teksleër bevat vier reëls met inligting oor 'n aansoeker vir 'n huislening in die volgende formaat:

<naam en van van aansoeker>

<maandelikse inkomste>

<maandelikse uitgawes>

<huisleningbedrag waarvoor aansoek gedoen is>

Inhoud van die drie teksleërs vir elk van die volgende aansoekers:  
 Abdul Brouwer, Paula Redhat en Xavier Outland

Abdul Brouwer	Paula Redhat	Xavier Outland
28000	25785	44741
12500	11190	20384
750000	888000	978000



Huisleningaansoek

Vind aansoeker

Evalueer aansoek-status

Jare

20

▼

Rente koers (%)

9.0

Evalueer aansoek

- Voltooi die kode soos wat dit in VRAAG 2.1 en VRAAG 2.2 beskryf word om funksionaliteit aan die program te gee.

2.1 Voltooi die kode in die huisleningklas **THuisLening/HuisLening** soos hieronder beskryf:

2.1.1 Skryf kode om die volgende vyf attribute by die klas te voeg:

Beskrywing	Name van attribute	
	Delphi	Java
Nam van aansoeker	fAansoekerNaam	aansoekerNaam
Besteembare inkomste	fBesteembareInkomste	besteembareInkomste
Huisleningbedrag waarvoor aansoek gedoen is	fLeningBedrag	leningBedrag
Getal jaar om huislening terug te betaal	fJare	jare
Rente koers	fRente koers	rente koers

(4)

2.1.2 Skryf kode om 'n konstruktor te skep wat die volgende parameterwaardes ontvang:

naam van aansoeker  
besteembare inkomste  
leningbedrag waarvoor aansoek gedoen is

Inisialiseer die relevante attribute deur die parameterwaardes te gebruik. Inisialiseer die attribute vir die getal jaar en die rentekoers na 0.

(5)



## AFDELING B

### VRAAG 2: OBJEK-GEORIENTEERDE PROGRAMMERING

<p>Kliënte wat eendomme wil koop, doen gewoonlik aansoek vir huislenings. Die volgende inligting oor aansoek wat ingedien is, word gewoonlik vereis:</p> <ul style="list-style-type: none"> <li>• Kwalifiseer die aansoeker vir 'n huislening?</li> <li>• Wat is die status van die aansoek, gebaseer op die duur van die lening in jaar en die rentekoers?</li> <li>• Wat is die bedrag van die paaiement wat elke maand terugbetaalbaar moet word?</li> </ul>
---

### INSTRUKSIES:

Delphi-programmeerders	Java-programmeerders
<ul style="list-style-type: none"> <li>• Die projek <b>Vraag2</b> is in die <b>Delphi-Vraag2_U.pas</b> hooftormleër met die naam <b>Vraag2_U.pas</b> <ul style="list-style-type: none"> <li>○ 'n Onvtooiide eenheidleër met die naam <b>HuisLening_U.pas</b></li> <li>○ 'n Tekstleër vir elke aansoeker</li> </ul> </li> <li>• Maak die onvtooiide projekleër met die naam <b>Vraag2_P.dpr</b> in die <b>Vraag2-leërgids</b> oop.</li> <li>• Besigtig ('View') (Ctrl+F12) die eenheidleër <b>HuisLening_U.pas</b> en voeg jou eksamennummer as kommentaar in die eerste reël van beide eenheidleërs <b>Vraag2_U.pas</b> en <b>HuisLening_U.pas</b> by.</li> </ul>	<ul style="list-style-type: none"> <li>• Die projek <b>Vraag2</b> is in die <b>Netbeans-Vraag2.java</b> -klasleër met die naam <b>Vraag2.java</b> <ul style="list-style-type: none"> <li>○ 'n Onvtooiide objekklas met die naam <b>HuisLening.java</b></li> <li>○ 'n Tekstleër vir elke aansoeker</li> </ul> </li> <li>• Maak die onvtooiide leër met die naam <b>Vraag2.java</b> en <b>Huislening.java</b> in die leërgids <b>Source Packages (src)</b> en <b>Vraag2Package</b> oop.</li> <li>• Voeg jou eksamennummer as kommentaar in die eerste reël van die klasse <b>Vraag2.java</b> en <b>HuisLening.java</b> by.</li> </ul>

Doen die volgende:

- Kompilleer en voer die program uit. Die program het tans geen funksionaliteit nie. Voorbeeld van die koppelvak (sien volgende bladsy):



Voorbeeld: Lys van geisers as die waarde 50 ingesleutel word:

Lys geisers	
QuickSun50	50
Solar Magic	
InHotWater	
WaterJoy 50	
Small Wonder	
Sun Magic	
Eco Wonder 50	

(6)

- Maak seker dat jou eksamennummer as kommentaar in die eerste reël van die program lêer ingesleutel is.
- Stoor jou program.
- In Drukstuk van die kode mag vereis word.

53 TOTAAL AFDELING A:



- As die huidige lesing gelyk is aan of groter is as die vorige lesing, gebruik die volgende tariewe om die bedrag verskuldig vir elektrisiteit wat gebruik is, te bereken:

Eenhede	Tariewe
0-600	R1,00 per eenheid
>600	R600 + R1,50 per eenheid vir alle eenhede wat 600 oorskry

- Vertoon die bedrag in die byskrif ('label') wat voorsien is, in 'n formaat wat 'n geldeenheid korrek tot twee desimale plekke insluit.

Voorbeeld:

Vorige lesing	2100
Huidige lesing	2735
Bereken bedrag verskuldig	
Bedrag verskuldig	R 652.50

### 1.3.2 [Knoppie – Lys geisers]

In 'n poging om elektrisiteit te bespaar word songeisers as 'n moonlikheid ondersoek. Drie groottes songeisers is beskikbaar op grond van hulle kapasiteit in liter. Daar word van die gebruiker verwag om een van die drie groottes in te sleutel: 50 (liter), 100 (liter) of 150 (liter).

'n Skikking wat stringe teks bevat wat elk 'n tipe geiser beskryf, word as deel van die gegewe kode voorsien.

Die beskrywing van elke tipe geiser begin met die kapasiteit van die geiser in liter, gevolg deur 'n aandagstreep en die handelsnaam van die geiser.

Voorbeeld: Die geiser wat as '50-Small Wonder' beskryf word, kan 50 liter water hou en die handelsnaam van die geiser is 'Small Wonder'.

Skryf kode om die tipes geisers te vertoon in die skikking wat ooreenstem met die kapasiteit wat deur die gebruiker ingesleutel is wanneer die **Lys geisers**-knoppie ('button') geklik word. Geen validering van die invoerwaarde word vereis nie.



(b) As die tipe opknapping **Teeëlwerk** is, moet 'n dialoogblokkie ('dialog box') gebruik word om die gebruiker toe te laat om die koste van teeëlwerk per vierkante meter in te sleutel. Voeg vyf vierkante meter by die vierkantemeter-waarde wat ingesleutel is om vir breekskade voorsiening te maak en bereken dan die koste. Vertoon die oppervlakte wat geteël moet word in vierkante meter. Die totale koste moet vertoon word, afgerond tot twee desimale plekke.

Voorbeeld: As 'n oppervlakte van  $100 \text{ m}^2$  ingesleutel is en die koste om te teël R50,00 per vierkante meter is, sal die volgende vertoon word:

Oppervlakte: 100 vierkante meter  
Totale koste: R 5250.00

(6)

### 1.3 Koppelvlak vir Water en elektrisiteit-oortjie ('tab')

Skryf kode om die volgende te doen:

#### 1.3.1 [Knoppie – Bereken bedrag verskuldig]

Die vorige lesing vir elektrisiteit wat gebruik is, is as 'n heelgetal in 'n teksblokkie ('text box') voorsien. Die gebruiker moet die huidige lesing as 'n heelgetal in 'n teksblokkie ('text box') insleutel.

- As die huidige lesing wat deur die gebruiker ingesleutel word, minder as die vorige lesing is, vertoon 'n geskikte boodskap en maak die teksblokkie ('text box') met die huidige lesing skoon.



## 1.2.2

**[Radioknoppie – Teëlwerk]**

As die Teëlwerk-radioknoppie ('radio button') geselekteer is, gebruik 'n veranderlike om die tipe opknapping as die karakter T te stoor.

(1)

## 1.2.3

**[Knoppie – Bereken en vertoon opknappingskoste]**

Die gebruiker moet die getal vierkante meter wat opgeknop moet word, as 'n heelgetal insleutel en die tipe opknapping wat gedoen moet word, kies. Die koste van elke tipe opknapping moet bereken word soos hieronder verduidelik.

(a) As die tipe opknapping **Verfwerk** is, gebruik die volgende inligting om te bereken hoeveel verf benodig word, asook die totale koste:

- Een liter verf word gebruik om 'n oppervlakte van agt vierkante meter te verf.
- Die prysstruktuur van verf per drom is soos volg:

- 1 liter-drom R 55.50
- 2 liter-drom R 92.30
- 5 liter-drom R 199.00

Vertoon die berekende waardes in die afvoerarea wat voorsien is en gebruik die volgende formaat:

Oppervlakte: <oppervlakte> vierkante meter  
Volume verf wat benodig word: <volume> liter

1 liter-drome: <getal 1 liter-drome>  
2 liter-drome: <getal 2 liter-drome>  
5 liter-drome: <getal 5 liter-drome>

Totale koste: <Totale koste van verf>

Die totale koste moet vertoon word, afgerond tot twee desimale plekke.

Voorbeeld: As die oppervlakte wat ingesleutel is, 148 m<sup>2</sup> is, sal die afvoer soos volg wees:

Oppervlakte: 148 vierkante meter  
Volume verf wat benodig word: 18.5 liter  
1 liter-drome: 0  
2 liter-drome: 2  
5 liter-drome: 3

Totale koste: R 781.60

(17)





- Vertoon die inligting van die advertensie in die afvoer-area wat voorsien is.

Voorbeeld van saamgestelde advertensies:

Voorbeeld 1: As die markwaarde van die eiendom R950 000 is, die verkooptys R850 000 is en die huis drie slaapkamers, twee badkamers en 'n swembad het, moet die advertensie soos volg vertoon:

Huis te koop:  
R850000#3Bed#2Bad#Swembad#Winskooop

Voorbeeld 2: As die markwaarde van die eiendom R750 000 is, die verkooptys R800 000 is en die huis twee slaapkamers, een badkamer en geen swembad het nie, moet die advertensie soos volg vertoon:

Huis te koop:  
R800000#2Bed#1Bad#

(11)

## 1.2 Koppelvlak van die Woonkamer-opknapping-oortjie ('tab')

Skryf kode om die volgende te doen:

### 1.2.1 [Radioknoppie – Verwerk]

As die Verwerk-radioknoppie ('radio button') geselekteer is, gebruik 'n veranderlike om die tipe opknapping as die karakter V te stoor.

(1)



## 1.1 Koppelvlak vir die Advertensie-oortjie ('tab')

Advertensie

Genereer advertensie

☐ Swembad

Markwaarde: 0

Verkoopprys: 0

Getal slaapkamers: 0

Getal badkamers: 0

**LET WEL:** Die **markwaarde** van 'n eiendom is die munisipale waarde waarvoor die eiendom gewaardeer is.

Die **verkoopprys** van 'n eiendom is die waarde van die verkoop van die eiendom.

**[Knoppie – Genereer advertensie]**

Skryf kode om die volgende te doen:

- Gebruik inligting uit die teksblokkies ('text boxes') wat voorsien is om 'n advertensie in die volgende formaat saam te stel:

Huis te koop:  
<verkoopprys>#<getal slaapkamers>slaap#<getal badkamers>Bad#

Gebruik die volgende inligting om die advertensie saam te stel:

- Die verkoopprys wat in die advertensie voorkom, moet met 'n geldeenheid geformateer wees. Moenie enige desimale waardes vertoon nie.

- As die Swembad-merklokke ('check box') geselekteer is, moet die teks 'Swembad#' by die advertensie se teks gevoeg word.

- As die verkoopprys van die eiendom minder as die markwaarde is, moet die woord 'Winskooop' by die advertensie se teks gevoeg word.



**SCENARIO**

Jy is by 'n maatskappy wat eiendomme koop, opknap en verkoop, as 'n sagteware-ontwikkeelaar in diens geneem.

**AFDELING A**

**VRAAG 1: ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

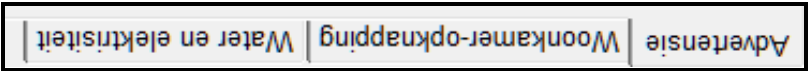
**INSTRUKSIES:**

Delphi-programmeerders	Java-programmeerders
<ul style="list-style-type: none"><li>Die projek <b>Vraag1</b> is in die <b>Delphi-</b> lêergids voorsien.</li><li>Maak die onvoltooide projekteer <b>Vraag1_P.dpr</b> in die <b>Vraag1</b>-lêergids oop.</li><li>Voeg jou eksamennummer as kommentaar in die eerste reël van die hoofvormeenheid <b>Vraag1_U.pas</b> by.</li></ul>	<ul style="list-style-type: none"><li>Die projek <b>Vraag1</b> is in die <b>Netbeans-</b> lêergids voorsien.</li><li>Maak die onvoltooide klas met die naam <b>Vraag1.java</b> in die lêergids <b>Source Packages (src)</b> en <b>Vraag1Package</b> oop.</li><li>Voeg jou eksamennummer as kommentaar in die eerste reël van die klas <b>Vraag1.java</b> by.</li></ul>

Doen die volgende:

- Kompileer en voer die program uit. Die koppelvlak vertoon drie verskillende oortjies ('tabs'), naamlik **Advertensie**, **Woonkamer-opknapping** en **Water en elektrisiteit**.

Voorbeeld van die oortjies ('tabs'):



- Voltooi die kode vir elke oortjie ('tab') soos wat dit in **VRAAG 1.1** tot **VRAAG 1.3** beskryf word.

**LET WEL:** **VRAAG 1.1** verwys na die **Advertensie-oortjie** ('tab').  
**VRAAG 1.2** verwys na die **Woonkamer-opknapping-oortjie** ('tab').  
**VRAAG 1.3** verwys na die **Water en elektrisiteit-oortjie** ('tab').



13. Die lêers wat jy benodig om hierdie vraestel te voltooi, is aan jou gegee op 'n disket/CD/DVD/geheuestokkie of op die skyfspasie wat aan jou toegeken is in die vorm van 'n wagwoordbeskermdde, uitvoerbare lêer:

- Delphi-leerders moet die lêer **DelphiDataAFR.exe** gebruik.
- Java-leerders moet die lêer **JavaDataAFR.exe** gebruik.

Doen die volgende:

- Dubbelklik op die lêer.
- Klik op die 'extract'-knoppie.
- Sleutel die volgende wagwoord in: **Properties87&%**

Wanneer die lêer onttrek is, sal die volgende lys van lêers in die **DelphiDataAFR/JavaDataAFR**- lêergids beskikbaar wees:

#### Delphi-lêers

**Vraag1:**  
Vraag1\_P.dpr  
Vraag1\_P.res  
Vraag1\_U.dfm  
Vraag1\_U.pas

**Vraag1:**  
meter1.png  
Vraag1.form  
Vraag1.java

**Vraag2:**  
Abdul Brouwer.txt  
HuisLening\_U.pas  
Paula Redhat.txt  
Vraag2\_P.dpr  
Vraag2\_P.res  
Vraag2\_U.dfm  
Vraag2\_U.pas  
Xavier Outland.txt

**Vraag2:**  
Abdul Brouwer.txt  
Paula Redhat.txt  
Xavier Outland.txt  
HuisLening.java  
Vraag2.form  
Vraag2.java

**Vraag3:**  
Vraag3\_P.dpr  
Vraag3\_P.res  
Vraag3\_U.dfm  
Vraag3\_U.pas

**Vraag3:**  
Vraag3.form  
Vraag3.java  
VuisKikkings.java



EASTERN CAPE

**INSTRUKSIES EN INLIGTING**

1. Hierdie vraestel is in DRIE afdelings verdeel. 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 kennis te neem 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 volgens die spesifikasies wat in elke vraag gegee word, beantwoord. Punte sal slegs volgens die voorgeskrewe vereistes toegeken word.
5. Beantwoord in elke vraag slegs wat gevra is. Byvoorbeeld, indien die vraag nie datavalidering vereis nie, sal geen punte vir datavalidering toegeken word nie.
6. Jou programme moet so 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. Roëtnes soos soek, sorteer en seleksie moet uit eerste beginsels ontwikkel word. Jy mag nie die ingeboude funksies van 'n programmeringstaal vir enige van hierdie roëtnes gebruik nie.
8. Alle datastrukture moet deur jou as programmeerder verklaar word. Jy mag nie komponente wat in die koppelvlak voorsien is, gebruik om data te stoor en later weer op te roep nie.
9. Jy moet jou werk gereeld stoor op die disket wat aan jou gegee is of op die skyfspasie wat aan jou vir hierdie eksamen toegeken is.
10. Maak seker dat jou eksamennummer 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 'n halfuur tyd vir drukwerk gegun word na die eksamensessie.
12. Aan die einde van hierdie eksamensessie moet jy 'n disket/CD/DVD/ gehuurstokkie met al jou werk daarop gestoor, inlewer OF jy moet seker maak dat jou werk op die skyfspasie wat vir hierdie eksamensessie aan jou toegeken is, gestoor is. Maak seker dat alle lêers gelees kan word.





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

**NASIONALE  
SENIOR SERTIFIKAAT**

**GRAAD 12**

**INLIGTINGSTEKNOLOGIE V1  
FEBRUARIE/MART 2015**

**PUNTE: 150**

**TYD: 3 uur**

**Hierdie vraestel bestaan uit 19 bladsye.**



Blaai om asseblief

Kopiereg voorbehou