MODULE NAME:	MODULE CODE:
PROGRAMMING 1B	PROG6112

ASSESSMENT TYPE:	TAKE-HOME EXAM (PAPER ONLY)
TOTAL MARK ALLOCATION:	120 MARKS
TOTAL TIME:	21 HOURS (midnight to 9PM on the same day)

By submitting this assessment, you acknowledge that you have read and understood all the rules as per the terms in the registration contract, in particular the assignment and assessment rules in The IIE Assessment Strategy and Policy (IIE009), the intellectual integrity and plagiarism rules in the Intellectual Integrity Policy (IIE023), as well as any rules and regulations published in the student portal.

INSTRUCTIONS:

1. *Please **adhere to all instructions**. These instructions are different from what is normally present, so take time to go through these carefully.*
2. ***Independent work is required**. Students are not allowed to work together on this assessment. Any contraventions of this will be handled as per disciplinary procedures in The IIE policy.*
3. ***No material may be copied from original sources, even if referenced correctly, unless it is a direct quote indicated with quotation marks.***
4. *All work must be adequately and correctly referenced.*
5. *You should paraphrase (use your own words) the concepts that you are referencing, rather than quoting directly.*
6. *Marks will be awarded for the quality of your paraphrasing.*
7. *This is an open-book assessment.*
8. *Assessments must be typed unless otherwise specified.*
9. ***Ensure that you save a copy of your responses.***
   - 9.1. *Complete your responses in a Word document.*
   - 9.2. *The document name must be your **name.student number.Module Code**.*
   - 9.3. *Once you have completed the assessment, upload your document under the **submission link** in the correct module in Learn.*

Additional instructions:

- *Calculators are allowed.*
- *Answer All Questions.*

1. ASSESSMENTS WRITTEN AT HOME:

a) No material may be copied from original sources, even if referenced correctly, unless it is a direct quote indicated with quotation marks.
b) All work must be adequately and correctly referenced.
c) You should paraphrase (use your own words) the concepts that you are referencing, rather than quoting directly.
d) This is an open-book assessment.
e) Assessments must be typed unless otherwise specified.
f) ***Ensure that you save a copy of your responses.***
  i. Complete your responses in a Word document.
  *ii. The document name must be your**.student number.Module Code**.*
  *iii. Once you have completed the assessment, upload your document under the **submission link** in the correct module in Learn.*

GENERAL REQUIREMENTS (Marks: 10)

Writing maintainable code in industry is vital. Therefore, you are generally required to:

- Follow good programming practices:
  - Variable types correct;
  - Variable scope correct; and
  - Class and method naming standards are correct.
- Insert comments to show logic and design for the support of code maintainability.
- Code efficiently. Redundant code must be avoided.
- Spend 15 minutes ensuring that your programs meet the general criteria below.

Requirement	**Maximum Mark**	**Examiner's Mark**	**Moderator's Mark**
Good Programming Practice • Not followed at all = 0 • Average – Minor changes required = 1 – 2 • Excellent – No changes necessary = 3	(3)		
Code Efficiency • Poor – Much code is duplicated = 0 • Average – Some redundant code = 1 • Excellent – Code very maintainable = 2	(2)		
Comment Statements • Poor – No comments = 0 • Average – Some comments = 1 – 2 • Excellent – All necessary comments were given to show logic = 3	(3)		
Program(s) compile and execute • No – Many changes required = 0 • Average – Minor changes required = 1 • Yes – No changes required = 2	(2)		
TOTAL MARKS	**10**		

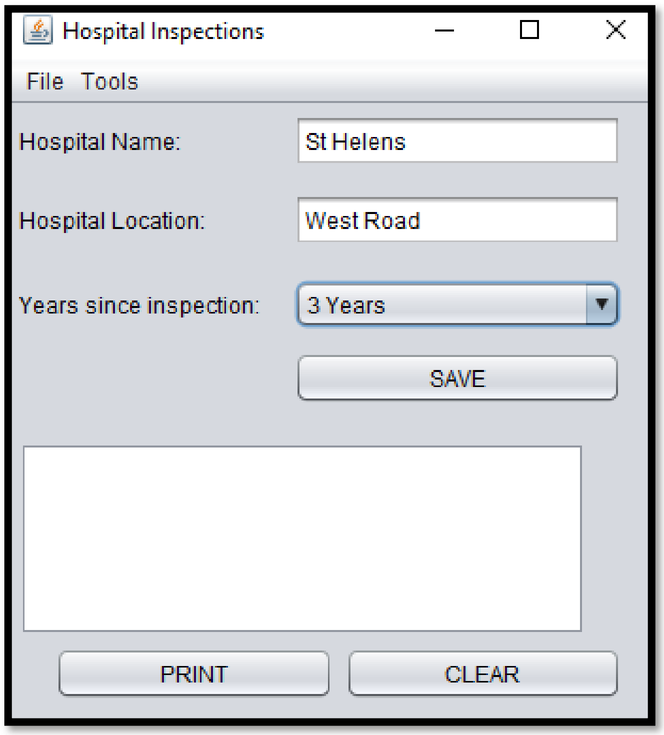
Question 1 **(Marks: 50)**

Develop a Java GUI application that will capture a maximum of five hospital inspection details, including the years since an inspection occurred.

Q.1.1

- Create two text fields on the form to allow users to enter the hospital name and location. Also, include a combo box that will be used to select the years since the last inspection. The year range is one year to five years.
- Also, add three buttons, one for saving the hospital inspection details to an array type of your choice, another button to print the hospital details from the array to a list box, and a third button to clear the controls.

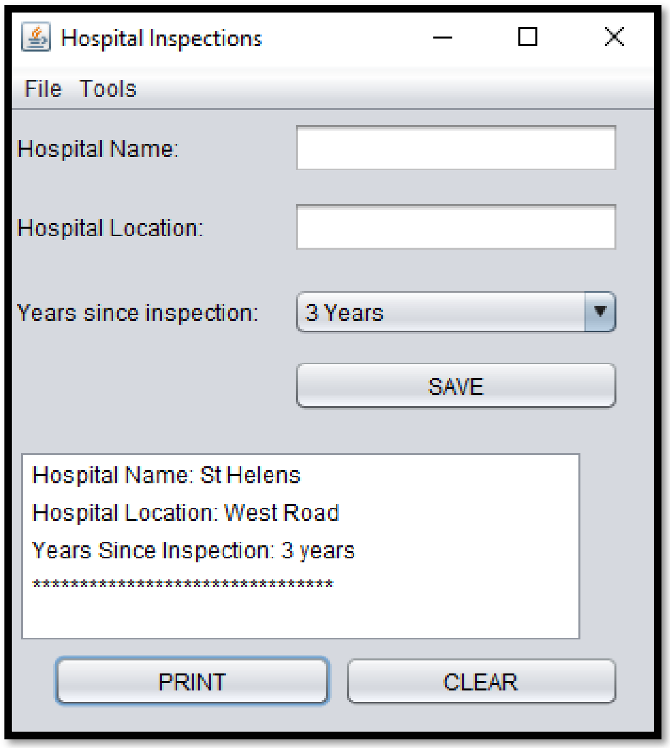
Sample screenshot



Hospital Name: St Helens

Hospital Location: West Road

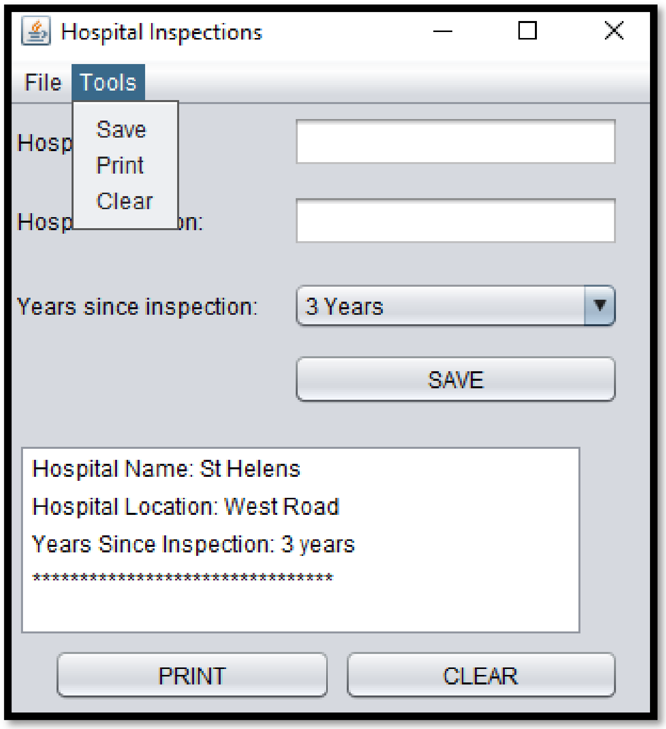
Years since inspection: 3 Years



```
Hospital Name: St Helens
Hospital Location: West Road
Years Since Inspection: 3 years
*********************************
```

Q.1.2 You are also required to create a menu system which will allow the user to exit the application under the file menu. Under the tools menu, allow the user to save the hospital inspections, print the inspections, and clear the controls. The layout of the form is left to your discretion. Marks will be allocated to the presentation and effectiveness of the layout, but the following layout is displayed for your convenience.

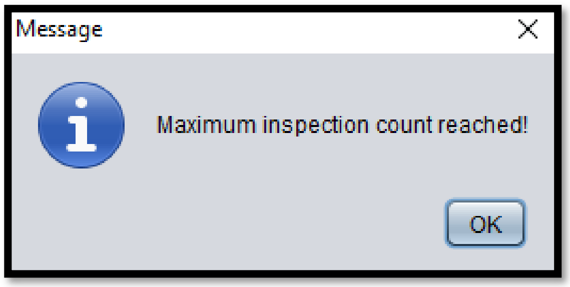
Sample screenshot



```
Hospital Name: St Helens
Hospital Location: West Road
Years Since Inspection: 3 years
*********************************
```

- If a user enters more than five hospital inspection details, display an error message.

Sample screenshot



Maximum inspection count reached!

Q.1.3	• Comment your program. • Marks will be allocated to the declaration of objects, logic, class definitions, method calls etc.

Question 1 Mark Allocation	Levels of Achievement				Feedback
	Excellent	Good	Developing	Poor	
	Score Ranges Per Level (½ marks possible)				
GUI form	**10** Correctly created without any errors.	**5-9** Correctly created with one or two minor errors.	**1-4** Created with many errors.	**0** Not created.	
Saves the hospital inspections to array(s)	**10** Correctly saves to array(s) without any errors.	**5-9** Correctly saves to array(s) with one or two minor errors.	**1-4** Saves to array(s) with many errors.	**0** Could not save to array(s).	
Prints the hospital inspections	**10** Correctly prints without any errors.	**5-9** Correctly prints with one or two minor errors.	**1-4** Prints with many errors.	**0** Could not print.	
File - Exit menu item	**5** Created and functions correctly without any errors.	**3-4** Created and functions correctly with one or two minor errors.	**1-2** Created and does not function correctly with many errors.	**0** Not created.	
Tools - Save menu	**4** Works the same as the save button.	**3** Works the same as the save button but has minor errors.	**1-2** Does not work the same as the save button and has a lot of errors.	**0** Was not created.	
Tools - Print menu	**4** Created and functions the same as the print button.	**3** Created and functions the same as the print button but has minor errors.	**1-2** Functions like the print button but has significant errors.	**0** Was not created.	
Tools - Clear menu	**4** Created and functions like the clear button.	**3** Functions like the clear button with minor errors.	**1-2** Functions like the clear button with significant errors.	**0** No menu was created.	

Error message	**3** Displayed if more than five (5) hospital inspection details were captured	**2** Displayed if more than five (5) hospital inspection details were captured with minor errors.	**1** Displayed if more than five (5) hospital inspection details were captured with major errors.	**0** No error message was displayed	

Question 2 **(Marks: 20)**

Write a Java application and use a two-dimensional array that will store three average property type prices in three different provinces. In your solution, include the total average property type price per province.

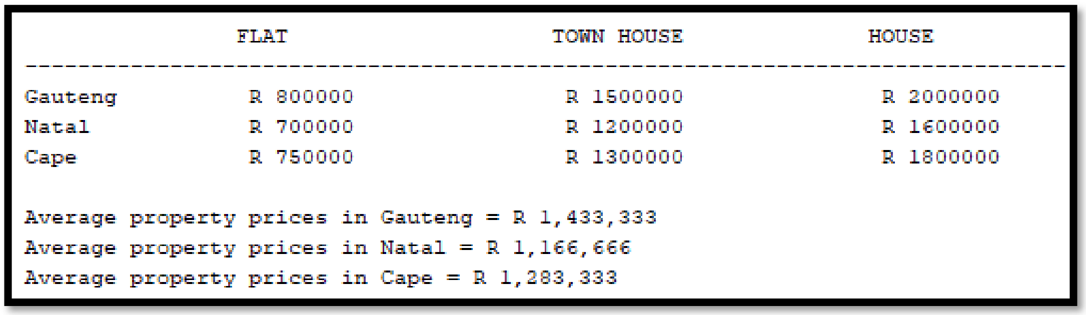
Your program must:

Q.2.1 contain a two-dimensional array to store three average property prices for three provinces.

PROVINCES		FLAT	TOWNHOUSE	HOUSE
Gauteng		R 800 000	R 1 500 000	R 2 000 000
Natal		R 700 000	R 1 200 000	R 1 600 000
Cape		R 750 000	R 1 300 000	R 1 800 000

Q.2.2 calculate and print the total average property type price for each province.

Sample screenshot

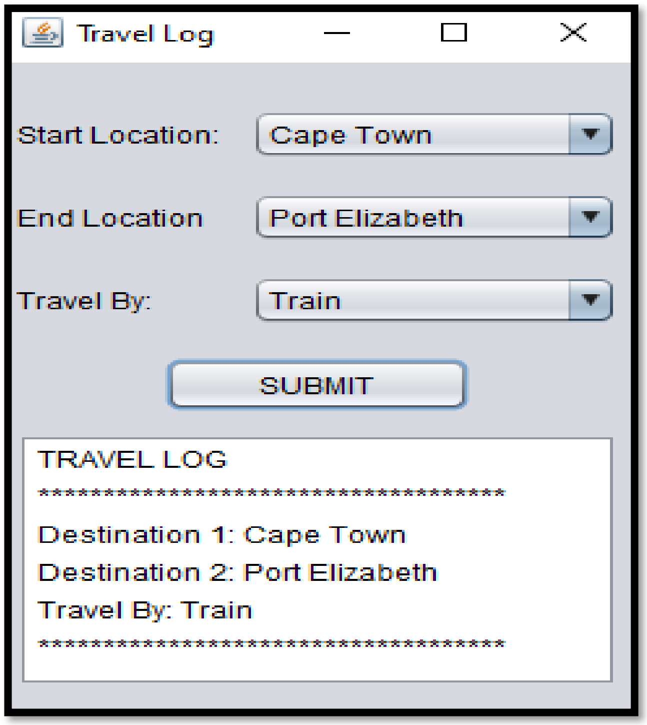


```
                  FLAT                     TOWN HOUSE             HOUSE
--------------------------------------------------------------------------------
Gauteng           R 800000                 R 1500000              R 2000000
Natal             R 700000                 R 1200000              R 1600000
Cape              R 750000                 R 1300000              R 1800000

Average property prices in Gauteng = R 1,433,333
Average property prices in Natal = R 1,166,666
Average property prices in Cape = R 1,283,333
```

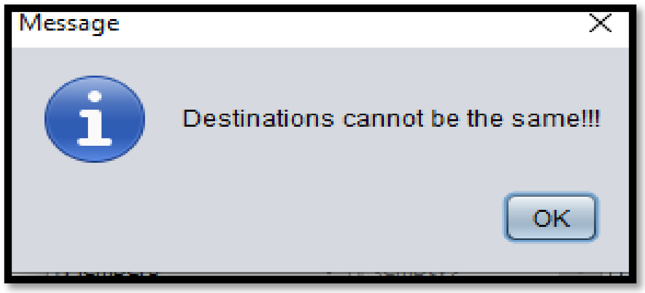
Question 2 Mark Allocation	**Levels of Achievement**				**Feedback**
	Excellent	**Good**	**Developing**	**Poor**	
	Score Ranges Per Level (½ marks possible)				
2D Array created to store the average property type prices	**5** Declared and populated without any errors.	**3-4** Declared and populated with one or two minor errors.	**1-2** Declared correctly, but the populated array had many errors.	**0** No declared.	
Single array to store the provinces	**5** Created without any errors.	**3-4** Created with one or two minor errors.	**1-2** Created with many errors.	**0** No single array to store.	
Calculate the total average property price per province	**5** Calculated without any errors.	**3-4** Calculated with one or two minor errors.	**1-2** Calculated with many errors.	**0** No calculations.	
Correct printing of the average property type prices, provinces and the total average prices	**5** Accurate printing without errors.	**3-4** Correct printing with one or two minor errors.	**1-2** Printing with many errors.	**0** No printing.	

Question 3		**(Marks: 40)**
Write a Java GUI application to capture travel information between two locations. The application must capture and display a user's travel location information.		
Q.3.1	Create three combo boxes and a list box to display the travel destination information on the form. Also, create a submit button that, when clicked, will capture the two locations.	
Q.3.2	When the submit button is clicked, save the start location, end location and the travel by information to a sequential file called travel.txt.	
Q.3.3	Load the data from the travel.txt file and populate the list box with the travel information.	
Q.3.4	The start and end location combo boxes must only contain three locations: • Cape Town • Durban • Port Elizabeth The travel by Combobox must only contain two travel types: • Airplane • Train	
Q.3.5	A user cannot travel to the same start and end location. A suitable error message must be displayed if this occurs.	
Sample screenshot		



```
TRAVEL LOG
****************************************
Destination 1: Cape Town
Destination 2: Port Elizabeth
Travel By: Train
****************************************
```

Sample screenshot if the same location is selected



Destinations cannot be the same!!!

Question 3 Mark Allocation	Levels of Achievement				Feedback
	Excellent	Good	Developing	Poor	
	Score Ranges Per Level (½ marks possible)				
GUI form created with all objects	**10** Created without any errors.	**5-9** Created with one or two minor errors.	**1-4** Created with many errors.	**0** Not created.	
Saving to the sequential file	**10** Saves without any errors.	**5-9** Saves with one or two errors.	**1-4**	**0** Does not save.	

			Saves but has many errors.		
Reading from the sequential file	**10** Correctly without any errors.	**5-9** Correctly with one or two errors.	**1-4** Many errors.	**0** Does not read.	
Populating the combo boxes with the correct values	**4** Correct values without any errors.	**3** Correct values with one or two errors.	**2-1** Many errors.	**0** Does not populate.	
Populating the list box with the travel information in the text file	**6** Populates without any errors.	**5-4** Populates with one or two errors.	**3-1** Populates with many errors.	**0** Does not populate.	

END OF PAPER