MODULE NAME:	MODULE CODE:
PROGRAMMING 1B	PROG6112

ASSESSMENT TYPE: ASSIGNMENT 1 (PAPER ONLY)

TOTAL MARK ALLOCATION: 100 MARKS

TOTAL HOURS: 15 HOURS

By submitting this assignment, 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. ***No material may be copied from original sources, even if referenced correctly, unless it is a direct quote indicated with quotation marks. No more than 10% of the assignment may consist of direct quotes.***
2. ***Make a copy of your assignment before handing it in.***
3. *Assignments must be typed unless otherwise specified.*
4. *All work must be adequately and correctly referenced.*
5. *Begin each section on a new page.*
6. *Follow all instructions on the assignment cover sheet.*
7. *This is an individual assignment.*

Referencing Rubric

Providing evidence based on valid and referenced academic sources is a fundamental educational principle and the cornerstone of high-quality academic work. Hence, The IIE considers it essential to develop the referencing skills of our students in our commitment to achieve high academic standards. Part of achieving these high standards is referencing in a way that is consistent, technically correct and congruent. This is not plagiarism, which is handled differently.

Poor quality formatting in your referencing will result in a penalty **of a maximum of ten percent being deducted from the percentage awarded**, according to the following guidelines. Please note, however, that **evidence of plagiarism in the form of copied or uncited work (not referenced), absent reference lists, or exceptionally poor referencing, may result in action being taken in accordance with The IIE's Intellectual Integrity Policy (0023)**.

Markers are required to provide feedback to students by indicating **(circling/underlining) the information that best describes the student's work.**

Minor technical referencing errors: 5% deduction from the overall percentage – the student's work contains **five or more errors** listed in the minor errors column in the table below.

Major technical referencing errors: 10% deduction from the overall percentage – the student's work contains **five or more errors** listed in the major errors column in the table below**.**

If both minor and major errors are indicated, then 10% only (and not 5% or 15%) is deducted from the overall percentage. The examples provided below are not exhaustive but are provided to illustrate the error.

Required: **Technically correct referencing style**	**Minor errors in technical correctness of referencing style** **Deduct 5% from percentage awarded**	**Major errors in technical correctness of referencing style** **Deduct 10% from percentage awarded**
Consistency • The same referencing format has been used for all in-text references and in the bibliography/reference list.	Minor inconsistencies. • The referencing style is generally consistent, but there are one or two changes in the format of in-text referencing and/or in the bibliography. • For example, page numbers for direct quotes (in-text) have been provided for one source, but not in another instance. Two book chapters (bibliography) have been referenced in the bibliography in two different formats.	Major inconsistencies. • Poor and inconsistent referencing style used in-text and/or in the bibliography/ reference list. • Multiple formats for the same type of referencing have been used. • For example, the format for direct quotes (in-text) and/or book chapters (bibliography/ reference list) is different across multiple instances.
Technical correctness Referencing format is technically correct throughout the submission. Position of the reference: a reference is directly associated with every concept or idea. For example, quotation marks, page numbers, years, etc. are applied correctly, sources in the bibliography/reference list are correctly presented.	**Generally, technically correct with some minor errors.** • The correct referencing format has been consistently used, but there are one or two errors. • Concepts and ideas are typically referenced, but a reference is missing from one small section of the work. • Position of the references: references are only given at the beginning or end of every paragraph. • For example, the student has incorrectly presented direct quotes (in-text) and/or book chapters (bibliography/reference list).	**Technically incorrect.** • The referencing format is incorrect. • Concepts and ideas are typically referenced, but a reference is missing from small sections of the work. • Position of the references: references are only given at the beginning or end of large sections of work. • For example, incorrect author information is provided, no year of publication is provided, quotation marks and/or page numbers for direct quotes missing, page numbers are provided for paraphrased material, the incorrect punctuation is used (in-text); the bibliography/reference list is not in alphabetical order, the incorrect format for a book chapter/journal article is used, information is missing e.g. no place of publication had been provided (bibliography); repeated sources on the reference list.
Congruence between in-text referencing and bibliography/ reference list • All sources are accurately reflected and are all accurately included in the bibliography/ reference list.	**Generally, congruence between the in-text referencing and the bibliography/ reference list with one or two errors.** • There is largely a match between the sources presented in-text and the bibliography. • For example, a source appears in the text, but not in the bibliography/ reference list or vice versa.	**A lack of congruence between the in-text referencing and the bibliography.** • No relationship/several incongruencies between the in-text referencing and the bibliography/reference list. • For example, sources are included in-text, but not in the bibliography and vice versa, a link, rather than the actual reference is provided in the bibliography.
In summary: the recording of references is accurate and complete.	In summary, at least **80%** of the sources are correctly reflected and included in a reference list.	In summary, at least **60%** of the sources are incorrectly reflected and/or not included in reference list.

Overall Feedback about the consistency, technical correctness and congruence between in-text referencing and bibliography:

.....

.....

STATE ALL ASSUMPTIONS you need to develop your answers. Your answers, code and screenshots must be saved in a single Microsoft Word document. Save this document as "PROG6112-Assignment–Student Number".

Question 1 **(Marks: 65)**

Extreme IT Products is a local supplier that specialises in the sales of the latest Information Technology hardware devices. The business has recently opened an outlet in the town you reside and has hired the software development house you work for to design a Java application to manage their products.

Your line manager has requested you to develop the application with the following requirements:

Q.1.1 When the application starts, it must display the following menu structure:

Sample Menu Screenshot

```
BRIGHT FUTURE TECHNOLOGIES APPLICATION
**************************************
Enter (1) to launch menu or any other key to exit
```

```
Please select one of the following menu items:
(1) Capture a new product.
(2) Search for a product.
(3) Update a product.
(4) Delete a product.
(5) Print report.
(6) Exit Application.
```

Q.1.2 If the user selects to capture a new product, you are required to save all the information supplied by the user into memory. To achieve this, you may use arrays or array lists.

Sample Capture Product Screenshot

```
CAPTURE A NEW PRODUCT
*************************
Enter the product code: A55
Enter the product name: EliteBook
Select the product category:
Desktop Computer - 1
Laptop - 2
Tablet - 3
Printer - 4
Gaming Console - 5

Product Category >> 2
Indicate the product warranty. Enter (1) for 6 months or any other key for 2 years. 2
Enter the price for EliteBook >> 15000
Enter the stock level for EliteBook >> 3
Enter the supplier for EliteBook >> IT_4_Africa
Product details has been saved successfully!!!
Enter (1) to launch menu or any other key to exit
```

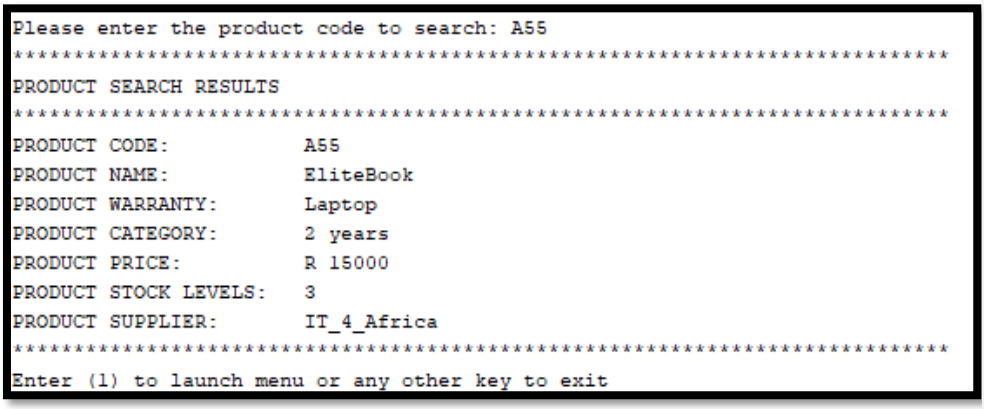
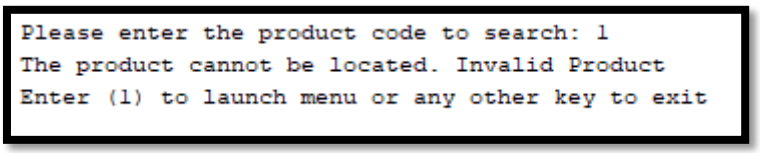
Q.1.4 Special consideration needs to be made for the selection of the product category. Allow the user to make a selection between the following categories:

- Desktop Computer.
- Laptop.
- Tablet.
- Printer.
- Gaming Console.

Q.1.5 If the user makes an incorrect product category selection, prompt the user to re-enter a valid product category.

Q.1.6 Provide the user with the option to select between the following product warranty options:

- 1 – Applies a six-month product warranty.
- Any other key applies a two-year warranty.

Q.1.7	Once the entire submission has been completed, the user must be informed that the product details have been successfully saved.	
Q.1.8	The user must have the ability to search for a product. The user will select menu item two (2), which will prompt the user to enter a product code. If a valid product is found in the application, then display the product details to the user. If no valid product is found, display an error message to the user that the product cannot be located. ***Sample Product Search Screenshot***  ***Sample Invalid Product Screenshot*** 	
Q.1.9	The user must have the option to delete a product that has been saved. The user must first enter the product code to be deleted. Next, the user must confirm whether they want to delete the product.	
Q.1.10	The user must also have the ability to update specific details of the product. For example, the user must first enter the product code and then confirm whether to update the following product details:	

- Update the product warranty.
- Update the product price.
- Update the product stock level.

Sample Update Product Screenshot

```
Please enter the product code to update: A55
Update the warranty? (y) Yes, (n) No n
Update the product price? (y) Yes, (n) No y
Enter the new price for EliteBook >> 14500
Update the stock level? (y) Yes, (n) No n
Product details has been updated successfully!!!
Enter (1) to launch menu or any other key to exit
```

Q.1.11 When the user selects to view a report, display the product report generated from the arrays in your application. You must create a class called **ReportData** which will contain **get and set methods for each item required in the report**.

Sample Report Screenshot

```
PRODUCT REPORT
================================================================================
PRODUCT 1
--------------------------------------------------------------------------------
PRODUCT CODE >>          A55
PRODUCT NAME >>          EliteBook
PRODUCT CATEGORY >>      2 years
PRODUCT WARRANTY >>      A55
PRODUCT PRICE >>         14500.0
PRODUCT LEVEL >>         3
PRODUCT SUPPLIER >>      IT_4_Africa
--------------------------------------------------------------------------------
PRODUCT 2
--------------------------------------------------------------------------------
PRODUCT CODE >>          A54
PRODUCT NAME >>          ExtremeBook
PRODUCT CATEGORY >>      2 years
PRODUCT WARRANTY >>      A54
PRODUCT PRICE >>         12500.0
PRODUCT LEVEL >>         3
PRODUCT SUPPLIER >>      IT_4_Africa
--------------------------------------------------------------------------------
PRODUCT 3
--------------------------------------------------------------------------------
PRODUCT CODE >>          A55
PRODUCT NAME >>          PS5
PRODUCT CATEGORY >>      2 years
PRODUCT WARRANTY >>      A55
PRODUCT PRICE >>         8500.0
PRODUCT LEVEL >>         3
PRODUCT SUPPLIER >>      Gaming_4_Africa
--------------------------------------------------------------------------------
================================================================================
TOTAL PRODUCT COUNT: 3
TOTAL PRODUCT VALUE: R 35500.0
AVERAGE PRODUCT VALUE: R 11833
================================================================================
Enter (1) to launch menu or any other key to exit
```

Q.1.12 Finally, provide the ability for the user to exit the application.

Additional Requirements:

1. In your solution, you must create a class called Products, which will contain all your working methods.
2. This class will, as a minimum, contain the following methods, but you are encouraged to add more methods:

    - SearchProduct();
    - SaveProduct();

- UpdateProduct();
- DeleteProduct();
- DisplayMenu();
- CaptureProduct();
- ExitApplication().

3. Finally, create a main class to run your application.

Question 2 (Marks: 35)

You have to design your own Java console application about any valid problem that your application must solve. Your solution can include solving a business problem, a new idea or even a game. Your application must make use of concepts such as arrays, loops, inheritance, constructors and information hiding. Output must be shown in the form of a report using the console.

In your solution, make use of as many concepts, and components dealt with in this course, but pay special attention to the following learning units:

- Learning Unit 1: Advanced arrays.
- Learning Unit 2: Introduction to inheritance.

Appendix A

Assessment Sheet (Marking Rubric)

MODULE NAME:	**MODULE CODE:**
PARENT MODULE NAME	**PROGRAMMING 1B**
CHILD MODULE NAME	**PROG6112**

STUDENT NAME:
STUDENT NUMBER:

Question 1 Mark Allocation	**Levels of Achievement**				**Feedback**
	Excellent	**Good**	**Developing**	**Poor**	
	Score Ranges Per Level (½ marks possible)				
Variables declared and assigned	**5**	**2 – 4**	**2**	**0 – 1**	
Menu and layout created	**6**	**3 – 5**	**2**	**0 – 1**	

Saving the captured values to memory	**6**	**4 – 5**	**2 – 3**	**0 – 1**	
Searching for a product	**6**	**4 – 5**	**2 – 3**	**0 – 1**	
Deleting a product	**6**	**4 – 5**	**2 – 3**	**0 – 1**	
Updating a product	**6**	**4 – 5**	**2 – 3**	**0 – 1**	

Product Report	**6**	**4 – 5**	**2 – 3**	**0 – 1**	
Report Data class containing get and set methods	**6**	**3 – 5**	**2**	**0 – 1**	
Products class created with working methods	**8**	**6 – 7**	**2 – 5**	**0 – 1**	
Good programming practise	**3**	**1 – 2**	**0 – 1**	**0**	

Code efficiency	**2**	**1**	**0 – 1**	**0**	
Program working	**2**	**1**	**0 – 1**	**0**	
Comments	**3**	**2**	**1 – 2**	**0**	

Question 2 Mark Allocation	Levels of Achievement				Feedback
	Excellent	Good	Developing	Poor	
	Score Ranges Per Level (½ marks possible)				
Variables declared	**4**	**2**	**0 – 1**	**0**	
Console application created with acceptable layout	**6**	**3 – 5**	**2**	**0 – 1**	
Concepts such as arrays, loops, inheritance, constructors and information hiding	**6**	**3 – 5**	**2**	**0 – 1**	

Code for application	**10**	**7 – 9**	**3 – 6**	**0 – 2**	
Creative application with more than basic functionality and effort	**4**	**2 – 3**	**1 – 2**	**0 – 1**	
File saved correctly	**1**	**0**	**0**	**0**	
Good programming practice	**1**	**0**	**0**	**0**	

Code efficiency	**1**	**0**	**0**	**0**	
Comments	**2**	**1**	**0 – 1**	**0**	