VAAL UNIVERSITY OF TECHNOLOGY
Inspiring thought. Shaping talent.

ASSIGNMENT 2 2025

FACULTY OF APPLIED & COMPUTER SCIENCES

DEPARTMENT OF COMPUTER SCIENCES

DIPLOMA: INFORMATION TEHNOLOGY

SUBJECT	:	DEVELOPMENT SOFTWARE 3.1
SUBJECT CODE	:	ASDSX3A
FORMATIVE ASSESSMENT	:	
DUE DATE	:	5 MAY 2024
DURATION	:	1 WEEEK
EXAMINER(S)	:	MRS S MOYO
MODERATOR(S)	:	MR N SOGANILE
REQUIREMENTS	**:**	**NOT APPLICABLE**

INSTRUCTIONS:

1. Download the zipped **YourStudentnumber_Assignment2_2025** NetBeans project that contains the GUI for the application to be coded.
2. Unzip the folder and rename the Project (replace the text **YourStudentnumber** with your student number)
3. Open the project from NetBeans and code the program according to the specifications on this question paper.
4. Where variables are declared, use the declared variables for uniformity.
5. When you are done coding the application, log onto VUTela and answer the online test that is based on this question paper.

MARKS:	**Total**	**=**	**100**
	Full marks	**=**	**100**

THE QUESTION PAPER CONISTS OF: 4 typed pages

Program Specification:

A **Java application** is required that can be used to sell products and store information about the sales in a relational database. Some classes to be used in this application have been partially coded for you, you need to code them to completion following the given specifications.

Follow the steps given to complete the required application:

1. Create a table named **tblproduct** in a database named **productsaledb**. **Figure 1** shows the structure of the table.

Field	Type	Null	Key
Barcode_No	int(8)	NO	PRI
Product_Name	varchar(18)	NO	
Product_Category	varchar(4)	NO	
Manufacturer	varchar(10)	YES	
Weight	int(11)	YES	
Unit_Price	int(11)	NO	
Amount_Paid	double	NO	

Figure 1

2. **Figure 2** shows some sales information in the database.

Barcode_No	Product_Name	Product_Category	Manufacturer	Weight	Unit_Price	Amount_Paid
16783234	Tuna	IBP	I & L	NULL	1500	17.25
17803064	Milk 1L	IBP	Clover	NULL	1500	17.25
26782561	Rump steak	WBP	NULL	250	12500	35.94
51343424	Bananas	WBP	NULL	450	4300	22.25
68900456	Boniato	WBP	NULL	446	4699	24.09

Figure 2

3. Code the Product class to include product category as one of the attributes.
4. Code the constructor of the product class to pass the **product category** attribute as a **third parameter**.
5. Create another parameterized constructor that passes the **total cost** attribute as parameter **number 5**.
6. Code the **toString** method of the Product class and its sub-classes to display the data as in Figure 3. Use get methods to access attributes.

Figure 3

==========

Data Access Class:

7. Create another class called **ProductDA** in the same package as the Product class and its sub-classes. Also create the exception classes necessary to handle any unexpected events that may disrupt the smooth running of the program.
8. Write the following methods in the ProductDA class:
   a. A method named **getProductSalesDbConnection**() to be used by the ProductDA class to establish a connection to the database productsaledb.
   b. A method named **addProduct(Product objProduct)**-to add any product to the database.
   c. A method named **returnAll()** to return all the rows stored in the database.
   d. A method named **calculateTotalWBP(),** calculates and returns the total amount for weight based products in the database.
   e. A method named **calculateTotalIBP**(), calculates and returns the total amount for item based products in the database.
   f. A method **updatePrice(**barcode number, percentage**)** to be used to update prices of products in the database. The method passes the barcode number as a parameter.
   g. A method **returnBarcodes()** to return barcodes only using an array list.

==========

Coding the GUI:

Create and code the GUI shown n Figure 4

9. Write code to establish the connection to the database when the form is created.

10. **Code the Sale button as follows:**
    -Use an if statement to check if radItemBasedProduct is selected, instantiate an object of the class and save it to the database,
    -otherwise instantiate an object of the WeightBased class and save it to the database -DO NOT DECLARE ANY VARIABLES.
    -Be sure to catch ALL exceptions that might be thrown.

11. **Code the View Sales button as follows:**
    -Call the **returnAll** method to display the data neatly on the text area (**see Figure 3**).
    -Be sure to catch ALL exceptions that might be thrown.

12. **Code the View Item Based button as follows:**
    -Call the returnAll method and display only item based products neatly on the text area .
    -Hint(declare a local array list(**arListProduct)** to store all the data returned by the method returnAll and then loop through this array list displaying item based products only.
    -Be sure to catch ALL exceptions that might be thrown.

13. **Code the updatePrice button as follows:**
    -write code to accept using an input box, the bar code number for the item whose price needs to be updated.
    write code to accept using an input box, the percentage (integer) to increase the price with.
    -call the method that updates prices.
    -Be sure to catch ALL exceptions that might be thrown.

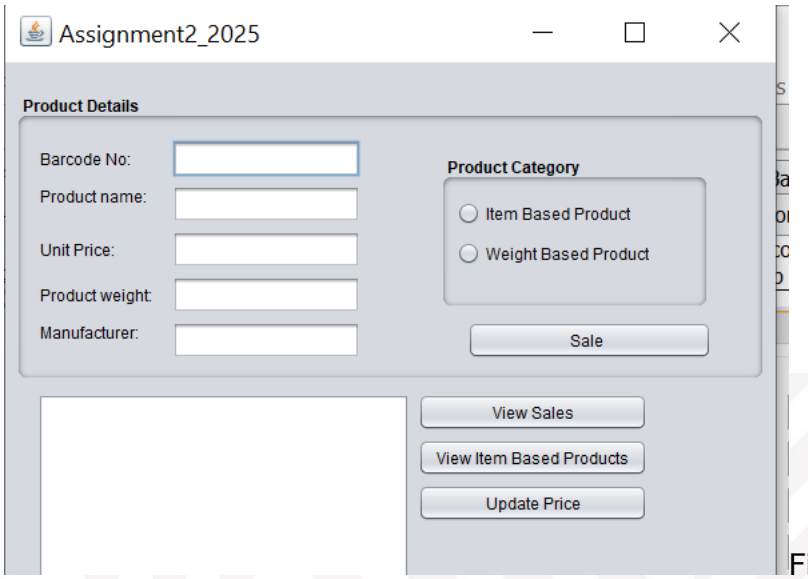

Figure 4

**********The End**********Good Luck**********