training solutions CAREER SUCCESS STARTS AT CTU

Faculty of Information Technology									
I declare that I am familiar with, and will abide to the Examination rules of CTU	**SUBJECT NAME:** **Beginner Java** **SUBJECT CODE:** **J521**								
	FORMATIVE ASSESSMENT **Duration**: Nov 14 – 15 Nov **Date** 15 November 2023 **Total Marks**: 100 **Total pages**: 13				**Examiner**: Junior Manganyi **Moderator:** Faith Muwishi				
[illegible] **Signature**	**Student number**								
	2	0	2	3	2	7	5	9	
	Surname: Poponi				**Initials**: M.P			/	%

Table of Contents

Summative Project Question(s)

Objectives:

Create Java application for a Product Management System that manages a collection of products using arrays or alternatively array lists.

Description: This project focuses on Java and data storage using data structures. Students will get a chance to work with Java's built-in classes and methods to create a functional Product Management System as a console application.

Requirements:

1. Create a `Product` class with the necessary properties and methods to represent a product.
2. Implement a `ProductManagement` class with methods for adding, editing, removing, and viewing products. Ensure that each product has a unique ID.
3. In the `Main` class, create an instance of the `ProductManagement` class and allow users to interact with the system by adding, editing, removing, and viewing products, including displaying the entered information for each product.
4. Ensure that the system supports the mentioned features and that the product data is stored and managed using arrays.
5. Your code should be well-structured, modular, and easy to understand.
6. Provide comments and documentation to explain the functionality of your code.

Expected Outcomes:

1. **User-Friendly Interface**: The application should provide a user-friendly interface that allows users to easily interact with the system.

2. **Add New Products**: Users should be able to add new products to the collection. Each product should have a unique ID assigned automatically. When a new product is added, it should be stored in the collection.

3. **Edit Existing Products**: Users should be able to edit and update the details of existing products, including the name, price, and quantity. Any changes made should be reflected in the product collection.

4. **Remove Products**: Users should have the ability to remove products from the collection. Upon removal, the product should no longer appear in the list.

5. **View Products**: Users should be able to view a list of all products in the collection. The information entered for each product, including the ProductID, Name, Price, and Quantity, should be displayed.

6. **Error Handling**: The application should handle potential errors gracefully. This includes providing clear error messages in case of unsupported inputs, user errors, or exceptions to prevent application crashes.

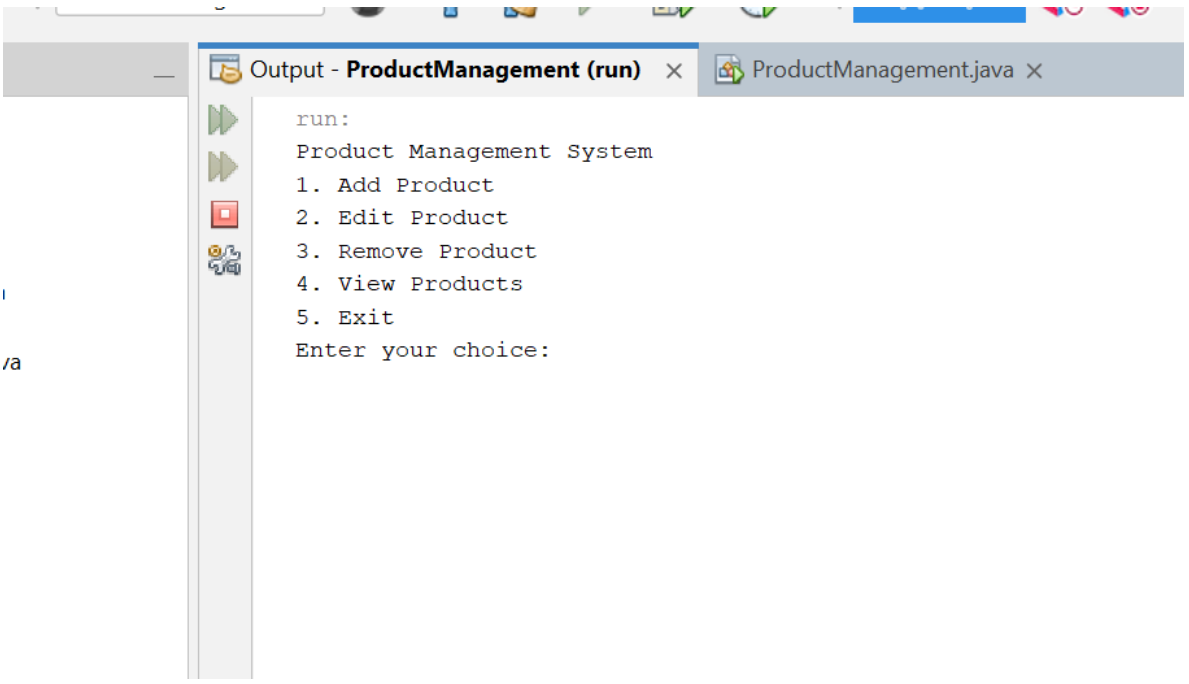
7. **Documentation**: The code should include comments explaining its functionality and purpose. Additionally, there should be comprehensive documentation, including a README file that describes the project's purpose, how to use the application, any known issues or limitations, and suggestions for future improvements.

Full code

```java
package productmanagement;

import java.util.ArrayList;
import java.util.Scanner;

// Product class to represent a product
class Product {
    private static int nextID = 1; // Static variable to generate unique IDs
    private int productID;
    private String name;
    private double price;
    private int quantity;

    // Constructor to initialize a product with a unique ID
    public Product(String name, double price, int quantity) {
        this.productID = nextID++;
        this.name = name;
        this.price = price;
        this.quantity = quantity;
    }

    // Getters and setters for product properties
    public int getProductID() {
        return productID;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public int getQuantity() {
        return quantity;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

    // Method to display product information
    public String toString() {
        return "Product ID: " + productID +
                "\nName: " + name +
                "\nPrice: R" + price +
                "\nQuantity: " + quantity;
    }
}

// ProductManagement class to manage products
public class ProductManagement {

    private ArrayList<Product> products = new ArrayList<>();

    // Method to add a new product to the collection
    public void addProduct(String name, double price, int quantity) {
        Product newProduct = new Product(name, price, quantity);
        products.add(newProduct);
        System.out.println("Product added successfully!\n" +
newProduct.toString());
    }

    // Method to edit an existing product
    public void editProduct(int productID, String name, double price,
int quantity) {
        for (Product product : products) {
            if (product.getProductID() == productID) {
                product.setName(name);
                product.setPrice(price);
                product.setQuantity(quantity);
                System.out.println("Product edited successfully!\n" +
product.toString());
                return;
            }
        }
        System.out.println("Product not found with ID: " + productID);
    }

    // Method to remove a product from the collection
    public void removeProduct(int productID) {
        products.removeIf(product -> product.getProductID() == productID);
        System.out.println("Product removed successfully!");
    }

    // Method to view all products in the collection
    public void viewProducts() {
        if (products.isEmpty()) {
            System.out.println("No products available.");
        } else {
            System.out.println("List of Products:");
            for (Product product : products) {
                System.out.println(product.toString() + "\n");
            }
        }
    }

    // Main class for user interaction

    public static void main(String[] args) {

        System.out.println("Product Management System");
         ProductManagement productManagement = new ProductManagement();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("1. Add Product\n2. Edit Product\n3. Remove Product\n4. View Products\n5. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter product name: ");
                    String name = scanner.next();
                    System.out.print("Enter product price: ");
                    double price = scanner.nextDouble();
                    System.out.print("Enter product quantity: ");
                    int quantity = scanner.nextInt();
                    productManagement.addProduct(name, price, quantity);
                    break;
                case 2:
                    System.out.print("Enter product ID to edit: ");
                    int editID = scanner.nextInt();
                    System.out.print("Enter new product name: ");
                    String editName = scanner.next();
                    System.out.print("Enter new product price: ");
                    double editPrice = scanner.nextDouble();
                    System.out.print("Enter new product quantity: ");
                    int editQuantity = scanner.nextInt();
                    productManagement.editProduct(editID, editName,
editPrice, editQuantity);
                    break;
                case 3:
                    System.out.print("Enter product ID to remove: ");
                    int removeID = scanner.nextInt();
                    productManagement.removeProduct(removeID);
                    break;
                case 4:
                    productManagement.viewProducts();
                    break;
                case 5:
                    System.out.println("Exiting the program.
Goodbye!");
                    scanner.close();
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please enter a
valid option.");
            }
        }
    }

}
```

User-friendly interface

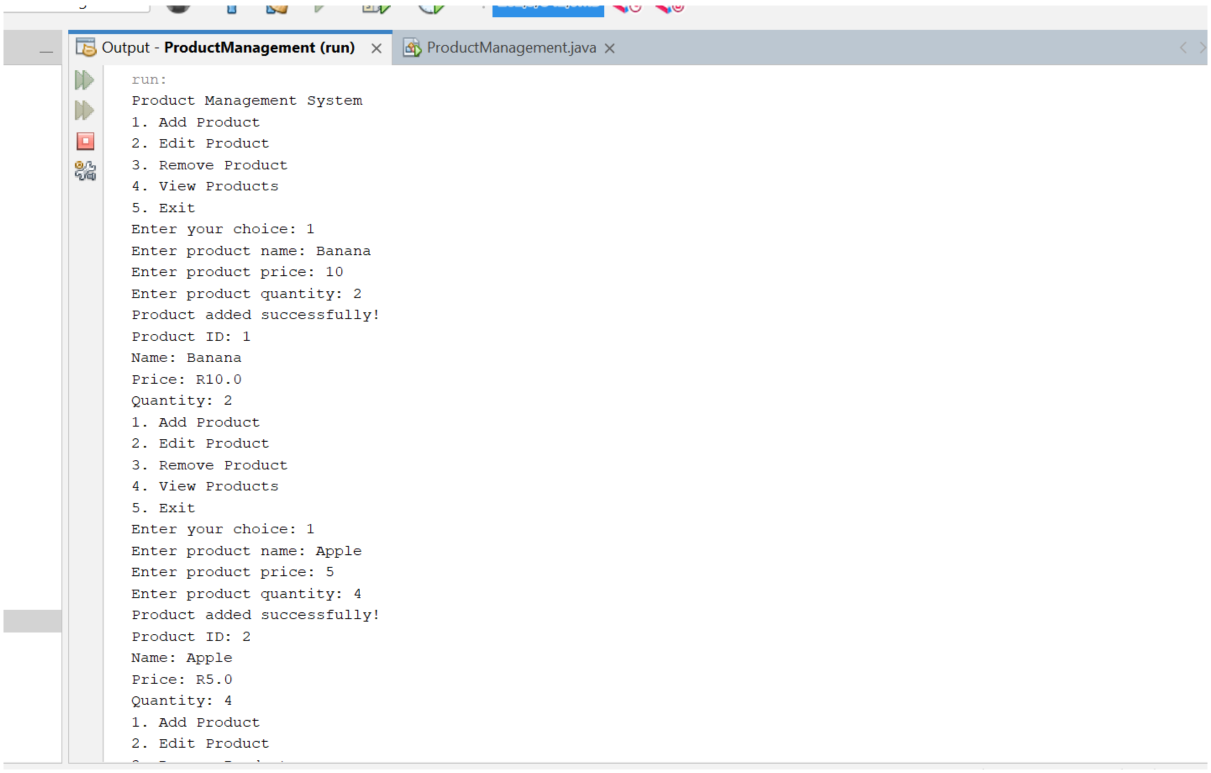
```
public static void main(String[] args) {

        System.out.println("Product Management System");
         ProductManagement productManagement = new ProductManagement();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("1. Add Product\n2. Edit Product\n3. Remove
Product\n4. View Products\n5. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter product name: ");
                    String name = scanner.next();
                    System.out.print("Enter product price: ");
                    double price = scanner.nextDouble();
                    System.out.print("Enter product quantity: ");
                    int quantity = scanner.nextInt();
                    productManagement.addProduct(name, price, quantity);
                    break;
                case 2:
                    System.out.print("Enter product ID to edit: ");
                    int editID = scanner.nextInt();
                    System.out.print("Enter new product name: ");
                    String editName = scanner.next();
                    System.out.print("Enter new product price: ");
                    double editPrice = scanner.nextDouble();
                    System.out.print("Enter new product quantity: ");
                    int editQuantity = scanner.nextInt();
                    productManagement.editProduct(editID, editName,
editPrice, editQuantity);
                    break;
                case 3:
                    System.out.print("Enter product ID to remove: ");
                    int removeID = scanner.nextInt();
                    productManagement.removeProduct(removeID);
                    break;
                case 4:
                    productManagement.viewProducts();
                    break;
                case 5:
                    System.out.println("Exiting the program. Goodbye!");
                    scanner.close();
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please enter a
valid option.");
            }
        }
    }

}
```



```
run:
Product Management System
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice:
```

Add New Products

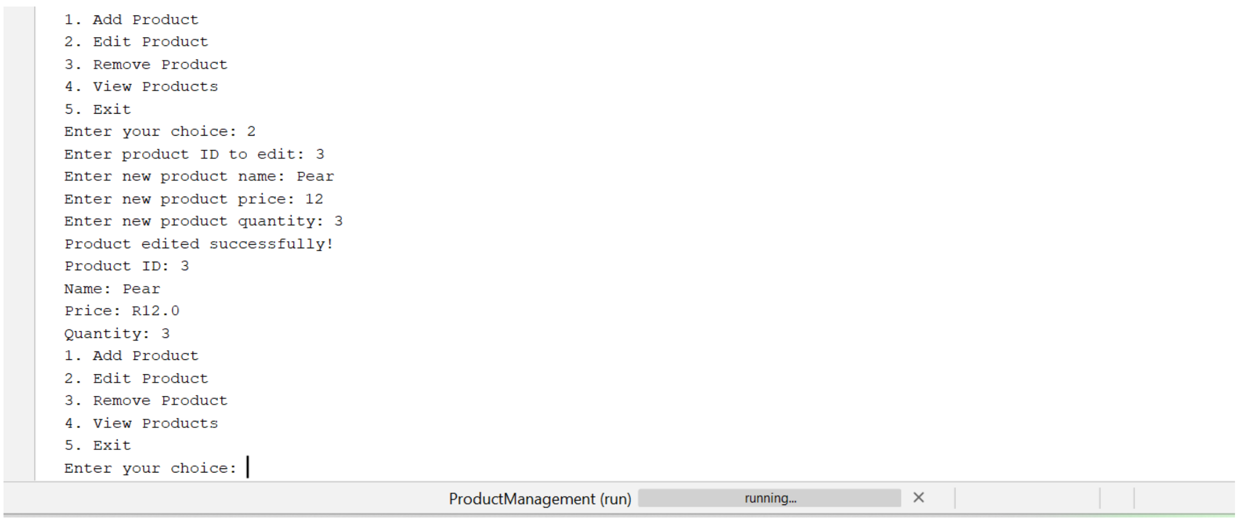
```
  public void addProduct(String name, double price, int quantity) {
         Product newProduct = new Product(name, price, quantity);
         products.add(newProduct);
         System.out.println("Product added successfully!\n" +
newProduct.toString());
     }
```



```
run:
Product Management System
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice: 1
Enter product name: Banana
Enter product price: 10
Enter product quantity: 2
Product added successfully!
Product ID: 1
Name: Banana
Price: R10.0
Quantity: 2
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice: 1
Enter product name: Apple
Enter product price: 5
Enter product quantity: 4
Product added successfully!
Product ID: 2
Name: Apple
Price: R5.0
Quantity: 4
1. Add Product
2. Edit Product
```

Edit Existing Product

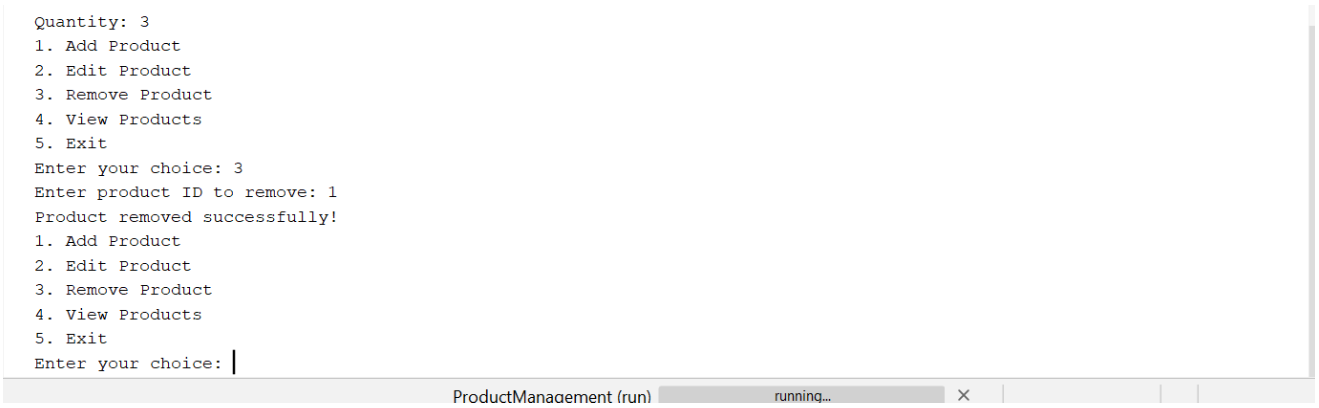
```
public void editProduct(int productID, String name, double price, int
quantity) {
        for (Product product : products) {
            if (product.getProductID() == productID) {
                product.setName(name);
                product.setPrice(price);
                product.setQuantity(quantity);
                System.out.println("Product edited successfully!\n" +
product.toString());
                return;
            }
        }
        System.out.println("Product not found with ID: " + productID);
    }
```



```
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice: 2
Enter product ID to edit: 3
Enter new product name: Pear
Enter new product price: 12
Enter new product quantity: 3
Product edited successfully!
Product ID: 3
Name: Pear
Price: R12.0
Quantity: 3
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice:
```

Remove Products

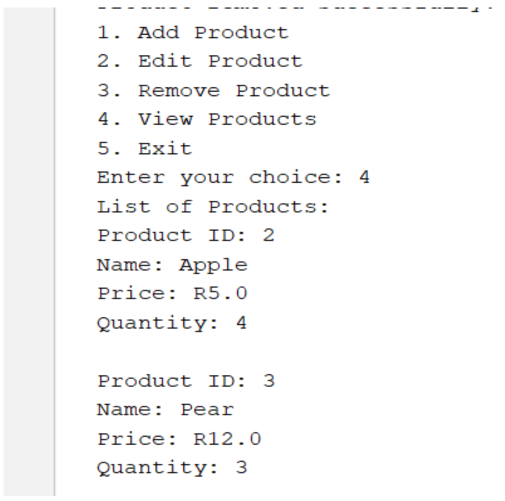
```java
public void removeProduct(int productID) {
        products.removeIf(product -> product.getProductID() == productID);
        System.out.println("Product removed successfully!");
    }
```



```
Quantity: 3
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice: 3
Enter product ID to remove: 1
Product removed successfully!
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice: |
```

View Products

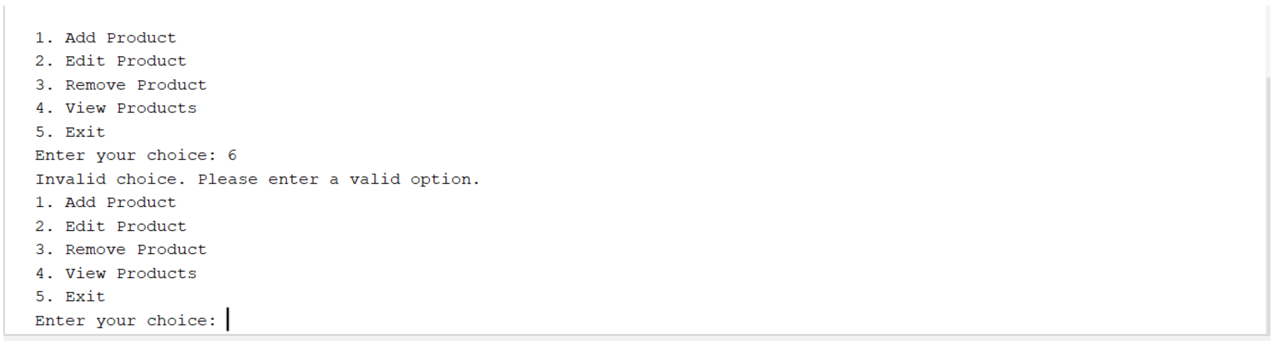
```java
public void viewProducts() {
        if (products.isEmpty()) {
            System.out.println("No products available.");
        } else {
            System.out.println("List of Products:");
            for (Product product : products) {
                System.out.println(product.toString() + "\n");
            }
        }
    }
```



```
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice: 4
List of Products:
Product ID: 2
Name: Apple
Price: R5.0
Quantity: 4

Product ID: 3
Name: Pear
Price: R12.0
Quantity: 3
```

Error Handling

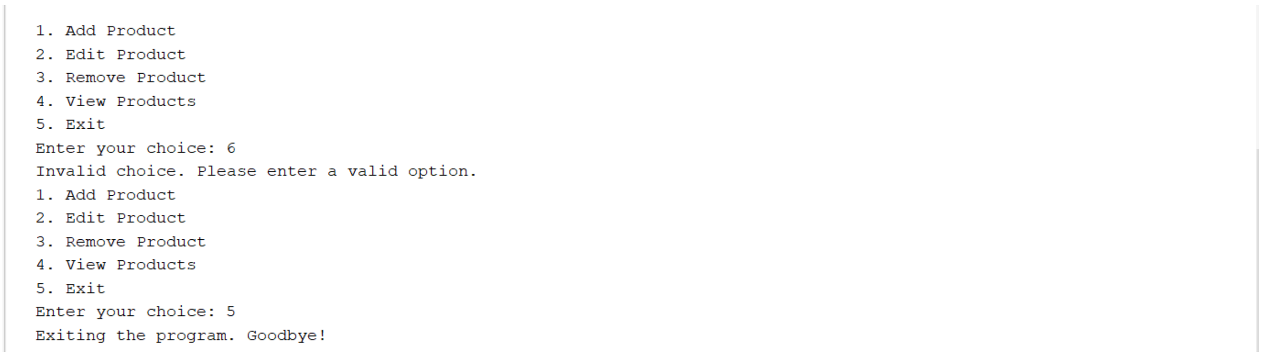
```
default:
                                System.out.println("Invalid choice. Please enter a valid option.");
```



```
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice: 6
Invalid choice. Please enter a valid option.
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice: 
```

Exit the program

```
System.out.println("Exiting the program. Goodbye!");
                                scanner.close();
                                System.exit(0);
```



```
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice: 6
Invalid choice. Please enter a valid option.
1. Add Product
2. Edit Product
3. Remove Product
4. View Products
5. Exit
Enter your choice: 5
Exiting the program. Goodbye!
```

DECLARATION OF AUTHENTICITY

I Mzukisi Poponi hereby

(FULL NAME)

declare that the contents of this exam for JD521 are entirely my work except the following elements: (List the elements of work in this project that were not self-generated as well as who the originator of the element is)

Element	Originator

Element Originator Signature: [illegible] Date: 15/11/2023