

**MSc Computer Science**

**School of Computing, Engineering and Digital technologies** ***Object-oriented Programming***

**CIS 4037-N**

**Academic Year:** 2023/24

**ICA Element 2:** *Test Cases* forStock Control Application with low stock reporting

**10 January 2024**

## **Class: ASCStockItem.java**

### **Test Case 1:** **Create ASC Stock Item**

Instantiate a Stock object with valid parameters. **STATUS: PASSED**

| ***Company*** | ***Product Title*** | ***Description*** | ***Unit Price (Pound)*** | ***Unit Price (Pence)*** | ***Quantity*** | ***Department*** |
| --- | --- | --- | --- | --- | --- | --- |
| ASC | Swimming Shorts | More reliable and water proof | 15 | 89 | 50 | SWM |

### 

### **Test Case 2: Validate Product Code Generation**

Check if the product code is generated automatically. **STATUS: PASSED**

| ***Company*** | ***Product Title*** | ***Description*** | ***Unit Price (Pound)*** | ***Unit Price (Pence)*** | ***Quantity*** | ***Department*** |
| --- | --- | --- | --- | --- | --- | --- |
| ASC | Running Sneakers | More reliable and easy to wear | 34 | 56 | 50 | RUN |

### **Test Case 3: Validate with random inputs**

Check if the input for the integers e.g jh for Quantity, Price it **STATUS: PASSED**

will show an alert on popup.

| ***Company*** | ***Product Title*** | ***Description*** | ***Unit Price (Pound)*** | ***Unit Price (Pence)*** | ***Quantity*** | ***Department*** |
| --- | --- | --- | --- | --- | --- | --- |
| ASC | Running Joggers | More reliable and easy to wear | 99 | 45 | 7 | CYC |

### 

### **Test Case 4: Validate with invalid inputs**

Check if the input for the integers e.g jh for Quantity, Price it **STATUS: PASSED**

will show an alert on popup. It is showing popup for enter valid

values.

| ***Company*** | ***Product Title*** | ***Description*** | ***Unit Price (Pound)*** | ***Unit Price (Pence)*** | ***Quantity*** | ***Department*** |
| --- | --- | --- | --- | --- | --- | --- |
| ASC | Running Joggers | More reliable and easy to wear | test | LL | 0IN | CYC |

### **Test Case 5: Validate CSV File Upload** **STATUS: PASSED**

Upload a CSV file with sales transaction data with the expected format separately for ASC despite the MSM records in the same table.

* **Values / Input:** Sample CSV file with multiple sales transactions.
* **EXPECTED OUTCOME:** Sales transactions recorded successfully.

### **Test Case 6: Validate Department-wise Product Code Formats**

Check if the product code format is generated correctly based **STATUS: PASSED**

on the department.

* **Values / Input:** Department ID: "2", Product Title: "Joggers", Product Description: "best for the long run", Unit Price (Pounds): 30,Unit Price (Pence): 79 Quantity: 10
* **EXPECTED OUTCOME:**  Product code follows the format for the RUN department.

### **Test Case 7: Create Sales Transaction Records on Sell Items**

Check Instantiate a Sales Transaction object with valid parameters **STATUS: PASSED**

for ASC items.

* **Values / Input:** Quantity Sold: 5
* **EXPECTED OUTCOME:**  Sales Transaction object created successfully.

### **Test Case 8: Invalid Inputs on Sold Quantity Form**

Check Instantiate a Sold Quantity with invalid parameters **STATUS: PASSED**

for ASC items. For example, Negative integer, strings etc,

It’ll show alert on popup to enter a valid input

* **Values / Input:** Quantity Sold: -6
* **EXPECTED OUTCOME:**  Please enter the valid input.

### **Test Case 9: Display All ASC Stock Records**

Retrieve and display all ASC stock records in proper format in **STATUS: PASSED**

the stock table.

* **Values / Input:** None
* **EXPECTED OUTCOME:**  List of all ASC stock records displayed.

### **Test Case 10: Display All ASC Sales Transactions**

Retrieve and display all ASC sales transactions recorded in **STATUS: PASSED**

Proper format in the stock table.

* **Values / Input:** None
* **EXPECTED OUTCOME:**  List of all ASC sales transactions displayed.

## 

## **Class: MSMStockItem.java**

### **Test Case 1:** **Create MSM Stock Item**

Instantiate a Stock object with valid parameters. **STATUS: PASSED**

| ***Company*** | ***Product Title*** | ***Description*** | ***Unit Price (Pound)*** | ***Unit Price (Pence)*** | ***Quantity*** | ***Department*** |
| --- | --- | --- | --- | --- | --- | --- |
| MSM | Swimming Shorts | More reliable and water proof | 15 | 89 | 50 | SWM |

### **Test Case 2: Validate Product Code Generation**

Check if the product code is generated automatically. **STATUS: PASSED**

| ***Company*** | ***Product Title*** | ***Description*** | ***Unit Price (Pound)*** | ***Unit Price (Pence)*** | ***Quantity*** | ***Department*** |
| --- | --- | --- | --- | --- | --- | --- |
| MSM | Running Sneakers | More reliable and easy to wear | 34 | 56 | 50 | RUN |

### **Test Case 3: Validate with random inputs**

Check if the input for the integers e.g jh for Quantity, Price it **STATUS: PASSED**

will show an alert on popup.

| ***Company*** | ***Product Title*** | ***Description*** | ***Unit Price (Pound)*** | ***Unit Price (Pence)*** | ***Quantity*** | ***Department*** |
| --- | --- | --- | --- | --- | --- | --- |
| MSM | Running Joggers | More reliable and easy to wear | 99 | 45 | 7 | CYC |

### 

### **Test Case 4: Validate with invalid inputs**

Check if the input for the integers e.g jh for Quantity, Price it **STATUS: PASSED**

will show an alert on popup. It is showing popup for enter valid

values.

**Values / Input:**

| ***Company*** | ***Product Title*** | ***Description*** | ***Unit Price (Pound)*** | ***Unit Price (Pence)*** | ***Quantity*** | ***Department*** |
| --- | --- | --- | --- | --- | --- | --- |
| MSM | Running Joggers | More reliable and easy to wear | twenty | Thirty two | fifty | CYC |

### **Test Case 5: Validate CSV File Upload** **STATUS: PASSED**

Upload a CSV file with sales transaction data with the expected

format separately for MSM despite the ASC records in the same

table.

* **Values / Input:** Sample CSV file with multiple sales transactions.
* **EXPECTED OUTCOME:** Sales transactions recorded successfully.

### **Test Case 6: Validate Department-wise Product Code Formats**

Check if the product code format is generated correctly based **STATUS: PASSED**

on the department.

* **Values / Input:** Department ID: "2", Product Title: "Joggers", Product Description: "best for the long run", Unit Price (Pounds): 30,Unit Price (Pence): 79 Quantity: 10
* **EXPECTED OUTCOME:**  Product code follows the format for the RUN department.

### 

### **Test Case 7: Create Sales Transaction Records on Sell Items**

Check Instantiate a Sales Transaction object with valid parameters **STATUS: PASSED**

for MSM items.

* **Values / Input:** Quantity Sold: 5
* **EXPECTED OUTCOME:**  Sales Transaction object created successfully.

### **Test Case 8: Invalid Inputs on Sold Quantity Form**

Check Instantiate a Sold Quantity with invalid parameters **STATUS: PASSED**

for MSM items. For example, Negative integer, strings etc,

It’ll show alert on popup to enter a valid input

* **Values / Input:** Quantity Sold: -6
* **EXPECTED OUTCOME:**  Please enter the valid input.

### **Test Case 9: Display All MSM Stock Records**

Retrieve and display all MSM stock records in proper format in **STATUS: PASSED**

the stock table.

* **Values / Input:** None
* **EXPECTED OUTCOME:**  List of all MSM stock records displayed.

### **Test Case 10: Display All MSM Sales Transactions**

Retrieve and display all MSM sales transactions recorded in **STATUS: PASSED**

Proper format in the stock table.

* **Values / Input:** None
* **EXPECTED OUTCOME:**  List of all MSM sales transactions displayed.

### **Low Stock Reporting**

### **Test Case 1: Report Low Stock**

Simulate a scenario where the stock quantity falls below the **STATUS: PASSED**

predefined threshold, triggering a low stock report.

On (Threshold: <5)

* **Values / Input:** Quantity Sold: 20
* **EXPECTED OUTCOME:**  Low stock alert generated.