Persistence

An Introduction to the CRUD Process

Produced by: Dr. Siobhán Drohan
Ms. Maireád Meagher

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/

Topic List

1. What is CRUD?
2. Recap of Shop V3.0
3. Shop V4.0 (Driver.java):
   – revised menu (making it CRUD compliant)

recap of case 1 (add a product)

recap of case 2 (list a product)

coding case 4 (delete a product)

coding case 3 (update a product)

CRUD

The four basic functions of **persistent storage**:

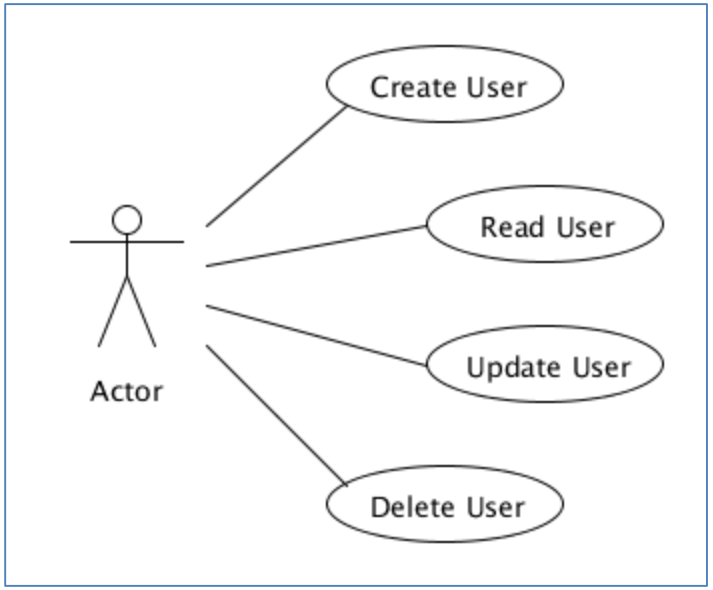
- Create or add new objects

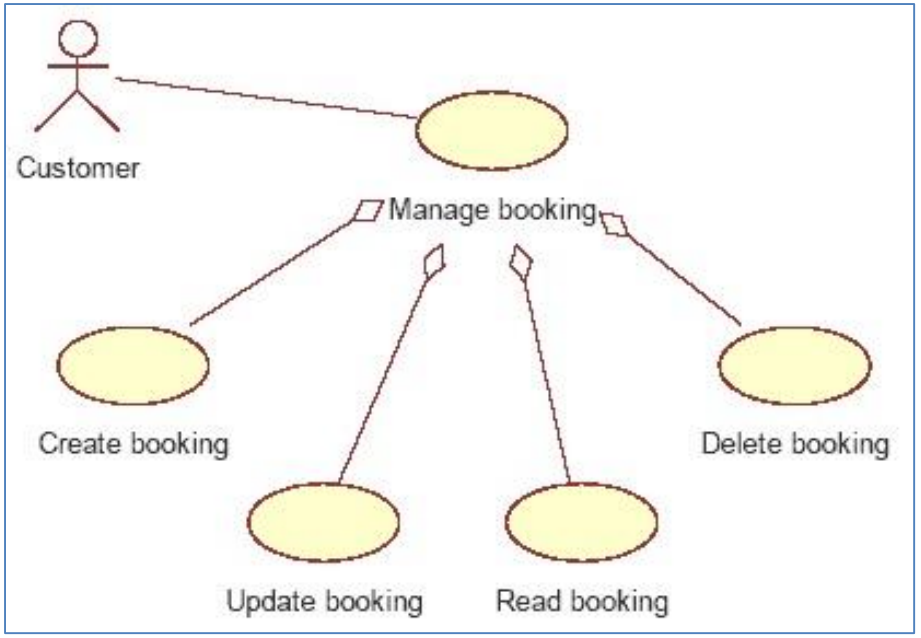
- Read, retrieve or search for existing objects

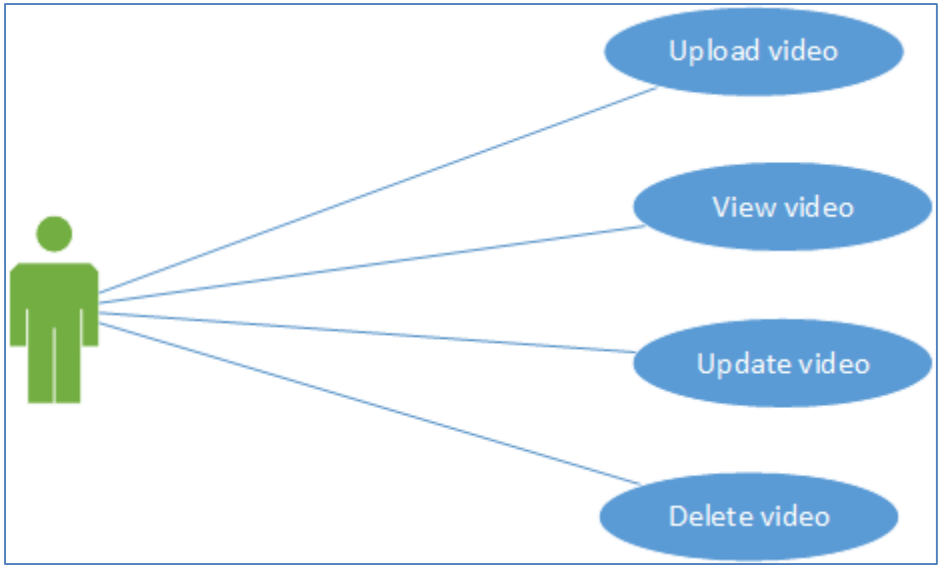
- Update or edit existing objects

- Delete existing objects

CRUD Examples







Topic List

1. What is CRUD?
2. Recap of Shop V3.0
3. Shop V4.0 (Driver.java):
   – revised menu (making it CRUD compliant)

recap of case 1 (add a product)

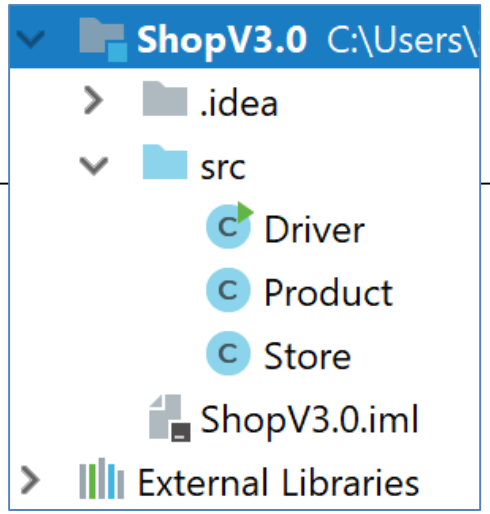
recap of case 2 (list a product)

coding case 4 (delete a product)

coding case 3 (update a product)

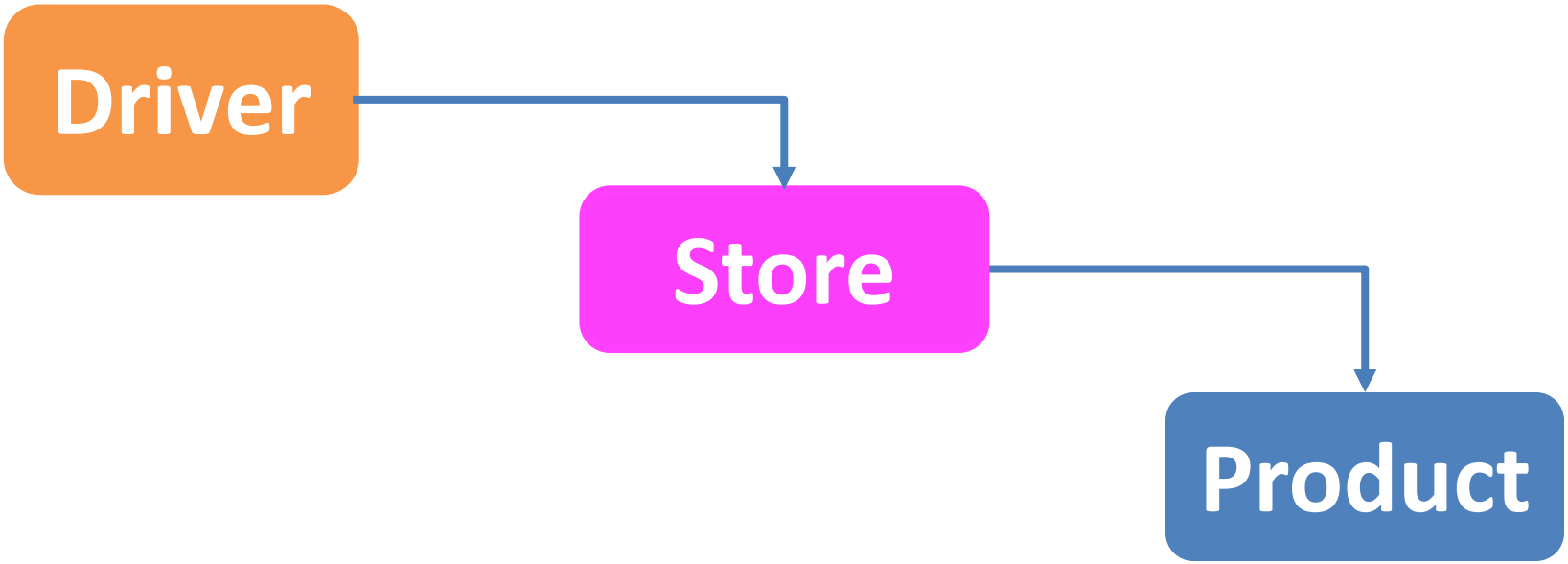
V3.0

Uses an **ArrayList of Products** to store the details.




```
Shop Menu
---------
  1) Add a product
  2) List the Products
  --------------------
  3) List the current products
  4) Display average product unit cost
  5) Display cheapest product
  6) List products that are more expensive than a given price
  --------------------
  0) Exit
==>>
```

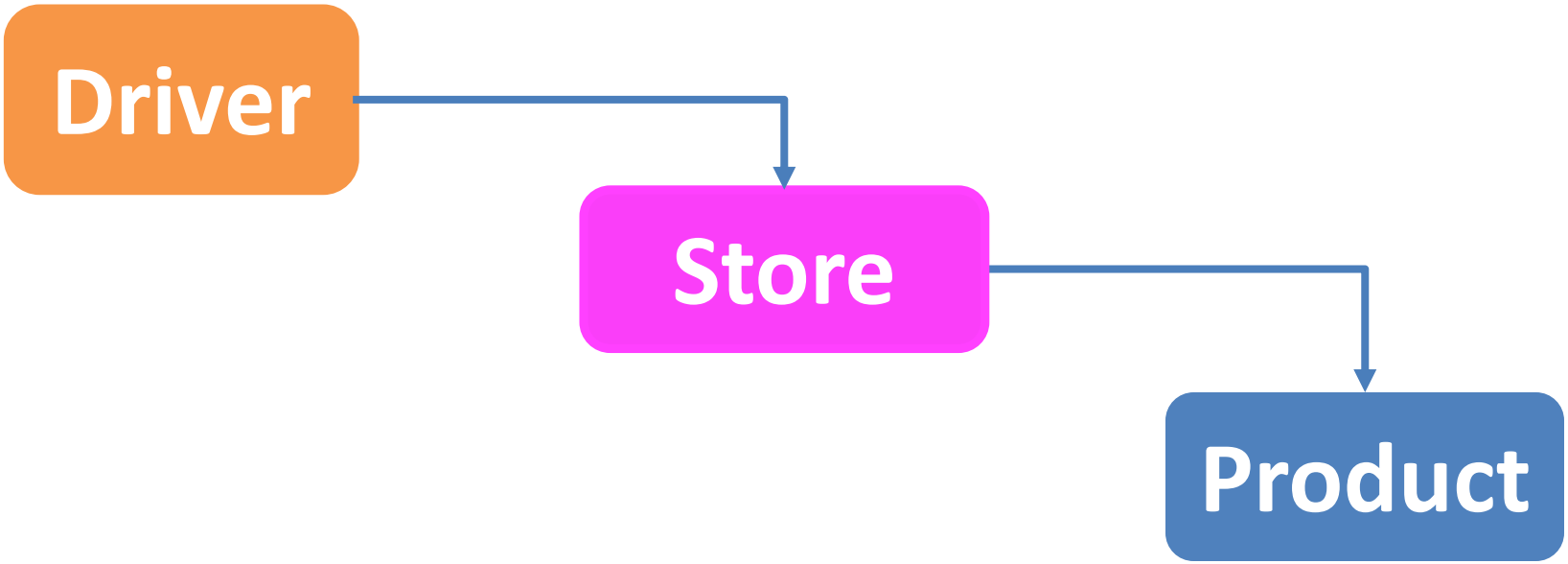
RECAP: Shop **V3.0**



Product class

- Four instance fields
  - product's name, code, unit cost, is in the current product line or not.
- Basic class with Constructors, Getters, Setters and toString methods

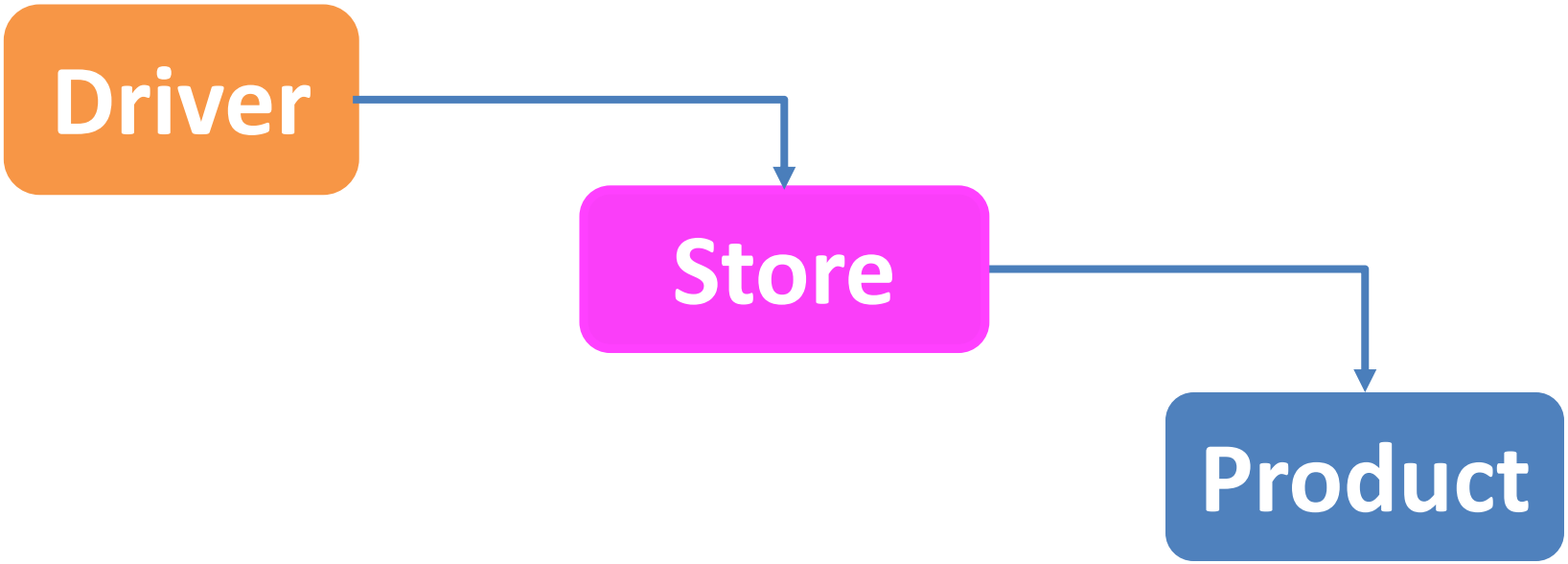
RECAP: Shop **V3.0**



Store class

- One instance field, **products** (an ***ArrayList of Product***).
- Many additional methods
  - listProducts(), cheapestProduct(), listCurrentProducts(), etc.

RECAP: Shop **V3.0**



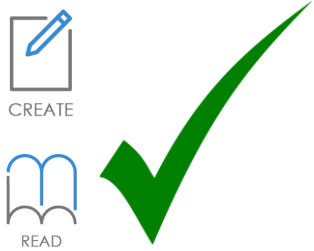
Driver

- Contains the **main()** method
- Runs the **menu**
- Negotiates with the user (i.e. handles **I/O**)

RECAP: Shop **V3.0**

Add Product: Menu Option 1.
Read a Product(s): Menu Options 2 - 6.

```
Shop Menu
---------
  1) Add a product
  2) List the Products
  --------------------
  3) List the current products
  4) Display average product unit cost
  5) Display cheapest product
  6) List products that are more expensive than a given price
  --------------------
  0) Exit
==>>
```



The menu has NO Update or Delete!

Topic List

1. What is CRUD?
2. Recap of Shop V3.0
3. Shop V4.0 (Driver.java):
   – revised menu (making it CRUD compliant)

recap of case 1 (add a product)

recap of case 2 (list a product)

coding case 4 (delete a product)

coding case 3 (update a product)

Shop V4.0 – Revised Menu

```
Shop Menu
---------
  1) Add a product
  2) List the products
  3) Update a product
  4) Delete a product
  --------------------
  5) List the current products
  6) Display average product unit cost
  7) Display cheapest product
  8) List products that are more expensive than a given price
  --------------------
  0) Exit
==>>
```

Option 1 – Create a Product
Option 2 – Read products
Option 3 – Update a product
Option 4 – Delete a product

Shop V4.0 – Revised Menu

```
private int mainMenu() {
    System.out.println("Shop Menu");
    System.out.println("---------");
    System.out.println("  1) Add a product");
    System.out.println("  2) List the products");
    System.out.println("  3) Update a product");
    System.out.println("  4) Delete a product");
    System.out.println("  --------------------");
    System.out.println("  5) List the current products");
    System.out.println("  6) Display average product unit cost");
    System.out.println("  7) Display cheapest product");
    System.out.println("  8) List products that are more expensive than a given price");
    System.out.println("  --------------------");
    System.out.println("  0) Exit");
    System.out.print("==>> ");
    int option = input.nextInt();
    return option;
}
```

Now we need to update the **switch** to:

- add code for case 3 (update) and 4 (delete) to Driver.java
- move the current options for 3-6 to be 5-8.

Shop V4.0 – Revised Menu

```
switch (option) {
    case 1:
        addProduct();
        break;
    case 2:
        printProduct();
        break;
    case 3:
        updateProduct();
        break;
    case 4:
        deleteProduct();
        break;
    case 5:
        printCurrentProducts();
        break;
    case 6:
        printAverageProductPrice();
        break;
    case 7:
        printCheapestProduct();
        break;
    case 8:
        printProductsAboveAPrice();
        break;
    default:
        System.out.println("Invalid option entered: " + option);
        break;
}
```

In Driver.java, we have provided a case 3 and 4, but we still need to write the actual methods:

- updateProduct
- deleteProduct)

We have moved the options for 3-6 to 5-8.

Topic List

1. What is CRUD?
2. Recap of Shop V3.0
3. Shop V4.0 (Driver.java):
   – revised menu (making it CRUD compliant)

recap of case 1 (add a product)

recap of case 2 (list a product)

coding case 4 (delete a product)

coding case 3 (update a product)

Driver.java code

```
switch (option) {
    case 1:
        addProduct();
        break;
    case 2:
        printProduct();
        break;
```

Code for case 1: Add a Product

```
//gather the product data from the user and create a new product.
private void addProduct() {
    //dummy read of String to clear the buffer - bug in Scanner class.
    input.nextLine();
    System.out.print("Enter the Product Name:  ");
    String productName = input.nextLine();
    System.out.print("Enter the Product Code:  ");
    int productCode = input.nextInt();
    System.out.print("Enter the Unit Cost:  ");
    double unitCost = input.nextDouble();
    System.out.print("Is this product in your current line (y/n): ");
    char currentProduct = input.next().charAt(0);
    boolean inCurrentProductLine = false;
    if ((currentProduct == 'y') || (currentProduct == 'Y'))
        inCurrentProductLine = true;

    store.add(new Product(productName, productCode, unitCost, inCurrentProductLine));
}
```

Topic List

1. What is CRUD?
2. Recap of Shop V3.0
3. Shop V4.0 (Driver.java):
   – revised menu (making it CRUD compliant)

recap of case 1 (add a product)

recap of case 2 (list a product)

coding case 4 (delete a product)

coding case 3 (update a product)

Driver.java code

```
switch (option) {
    case 1:
        addProduct();
        break;
    case 2:
        printProduct();
        break;
```

Code for case 2: List the Products

```
private void printProduct() {
    System.out.println("List of Products are:");
    System.out.println(store.listProducts());
}
```

Driver.java code

```
switch (option) {
    case 1:
        addProduct();
        break;
    case 2:
        printProduct();
        break;
```

Code for case 2: List the Products

```
private void printProduct() {
    System.out.println("List of Products are:");
    System.out.println(store.listProducts());
}
```

Output from case 2 call:

```
==>> 2
List of Products are:
0: Product description: tv, product code: 1234, unit cost: 349.99, currently in product line: true
1: Product description: phone, product code: 2345, unit cost: 299.99, currently in product line: true
2: Product description: amazon echo, product code: 4543, unit cost: 89.0, currently in product line: false
```

Code for case 2: List the Products

Store.java code

```
public String listProducts(){
    if (products.size() == 0){
        return "No products";
    }
    else{
        String listOfProducts = "";
        for (int i = 0; i < products.size(); i++){
            listOfProducts = listOfProducts + i + ": " + products.get(i) + "\n";
        }
        return listOfProducts;
    }
}
```

Topic List

1. What is CRUD?
2. Recap of Shop V3.0
3. Shop V4.0 (Driver.java):
   – revised menu (making it CRUD compliant)

recap of case 1 (add a product)

recap of case 2 (list a product)

coding case 4 (delete a product)

coding case 3 (update a product)

Driver.java code

```
switch (option) {
    case 1:
        addProduct();
        break;
    case 2:
        printProduct();
        break;
    case 4:
        deleteProduct();
        break;
```

Code for case 4: Delete a Product

```
private void deleteProduct(){
    //list the products and ask the user to choose the product to delete
    System.out.println(store.listProducts());
    System.out.print("Enter the index of the product to delete ==> ");
    int index = input.nextInt();

    //delete the product at the given index
    store.getProducts().remove(index);
    System.out.println("Product deleted.");
```

The deleteProduct() method does not have any **validation**:

- What happens if there are **no products** in the ArrayList?
- What happens if the **index number does not exist** in the ArrayList?

```
private void deleteProduct(){
    //list the products and ask the user to choose the product to delete
    System.out.println(store.listProducts());
    System.out.print("Enter the index of the product to delete ==> ");
    int index = input.nextInt();

    //delete the product at the given index
    store.getProducts().remove(index);
    System.out.println("Product deleted.");
```

Validation:

- Only process the delete if **there are products** in the ArrayList and the **number entered is less than the size** of the ArrayList.

```
private void deleteProduct(){
    //list the products
    System.out.println(store.listProducts());

    if (store.getProducts().size() > 0) {
        //only ask the user to choose the product to delete if products exist
        System.out.print("Enter the index of the product to delete ==> ");
        int index = input.nextInt();

        if ((index >= 0) && (index < store.getProducts().size())) {
            //if the index is valid, delete the product at the given index
            store.getProducts().remove(index);
            System.out.println("Product deleted.");
        }
        else{
            System.out.println("There is no product for this index number");
        }
    }
```

Topic List

1. What is CRUD?
2. Recap of Shop V3.0
3. Shop V4.0 (Driver.java):
   – revised menu (making it CRUD compliant)

recap of case 1 (add a product)

recap of case 2 (list a product)

coding case 4 (delete a product)

coding case 3 (update a product)

Coding case 3: Updating a Product

Driver.java code

```
switch (option) {
    case 1:
        addProduct();
        break;
    case 2:
        printProduct();
        break;
    case 3:
        updateProduct();
        break;
    case 4:
        deleteProduct();
        break;
    case 5:
        printCurrentProducts();
        break;
```

Coding case 3: Updating a Product

```java
private void updateProduct(){
    //list the products
    System.out.println(store.listProducts());

    //ask the user to choose a product
    System.out.print("Enter the index of the product to update ==> ");
    int index = input.nextInt();

    //gather new details for each field from the user
    input.nextLine();  //dummy read of String to clear buffer - bug in Scanner.
    System.out.print("Enter the Product Name:  ");
    String productName = input.nextLine();
    System.out.print("Enter the Product Code:  ");
    int productCode = input.nextInt();
    System.out.print("Enter the Unit Cost:  ");
    double unitCost = input.nextDouble();
    System.out.print("Is this product in your current line (y/n): ");
    char currentProduct = input.next().charAt(0);
    boolean inCurrentProductLine = false;
    if ((currentProduct == 'y') || (currentProduct == 'Y'))
        inCurrentProductLine = true;

    //retrieve the selected product from the ArrayList and update the details
    Product product = store.getProducts().get(index);
    product.setProductCode(productCode);
    product.setProductName(productName);
    product.setUnitCost(unitCost);
    product.setInCurrentProductLine(inCurrentProductLine);
}
```

Driver.java

The editProduct() method does not have any **validation** in it:

- What happens if there are **no products** in the ArrayList?
- What happens if the **index number does not exist** in the ArrayList?

Coding case 3: Updating a Product

Coding case 3: Updating a Product

```
private void updateProduct(){
    //list the products
    System.out.println(store.listProducts());

    if (store.getProducts().size() > 0) {
        //only ask the user to choose a product if products exist
        System.out.print("Enter the index of the product to update ==> ");
        int index = input.nextInt();

        if ((index >= 0) && (index < store.getProducts().size())) {
            //if the index is valid, gather new details for each field from the user
            input.nextLine();  //dummy read of String to clear buffer - bug in Scanner.
            System.out.print("Enter the Product Name:  ");
            String productName = input.nextLine();
            System.out.print("Enter the Product Code:  ");
            int productCode = input.nextInt();
            System.out.print("Enter the Unit Cost:  ");
            double unitCost = input.nextDouble();
            System.out.print("Is this product in your current line (y/n): ");
            char currentProduct = input.next().charAt(0);
            boolean inCurrentProductLine = false;
            if ((currentProduct == 'y') || (currentProduct == 'Y'))
                inCurrentProductLine = true;

            //retrieve the selected product from the ArrayList and update the details
            Product product = store.getProducts().get(index);
            product.setProductCode(productCode);
            product.setProductName(productName);
            product.setUnitCost(unitCost);
            product.setInCurrentProductLine(inCurrentProductLine);
        }
        else {
            System.out.println("There are no products for this index number");
        }
    }
}
```

Driver.java

Any Questions?