Input / Output

Scanner class

Produced by: Dr. Siobhán Drohan
Mr. Colm Dunphy
Mr. Diarmuid O'Connor
Dr. Frank Walsh

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

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

Input in Java: the **Scanner** Class

- The **Scanner** class comes with Java.
- It allows us to **take in data from the console** / terminal window.
- It is part of the **java.util** package in the Java Application Programming Interfaces (API).

Input in Java: the **Scanner** Class

- In order to use the Scanner class,
  place the following line as the **first line of code in your file**
  (i.e. before class declaration):

```
import java.util.Scanner;
```

Driver.java | Product.java

```
1  import java.util.Scanner;
2
3  /**
4   * This class runs the application and handles the Product I/O
5   * @version 1.0
6   */
7  public class Driver{
8
```

Input in Java: the **Scanner** Class

- Having imported the util package, you will need to write the following instruction in your program.

```
Scanner input = new Scanner(System.in);
```

Driver.java | Product.java

```
1  import java.util.Scanner;
2
3  /**
4   * This class runs the application and handles the Product I/O
5   * @version 1.0
6   */
7  public class Driver{
8
9      private Scanner input = new Scanner(System.in);
```

Input in Java: the **Scanner** Class

- This declares a Scanner **object** called **input** (you can name this object anything you wish).
- You must have this instruction to be able to call the methods in the Scanner class.

Driver.java | Product.java

```
1  import java.util.Scanner;
2
3  /**
4   * This class runs the application and handles the Product I/O
5   * @version 1.0
6   */
7  public class Driver{
8
9      private Scanner input = new Scanner(System.in);
```

Input in Java: the **Scanner** Class

- Now that a Scanner object is set up,
  we can use all the **input method**s
  that have been defined in the Scanner class.

- There are **methods** to take in:
  - ints,
  - doubles,
  - Strings,
  - chars,
  - etc.

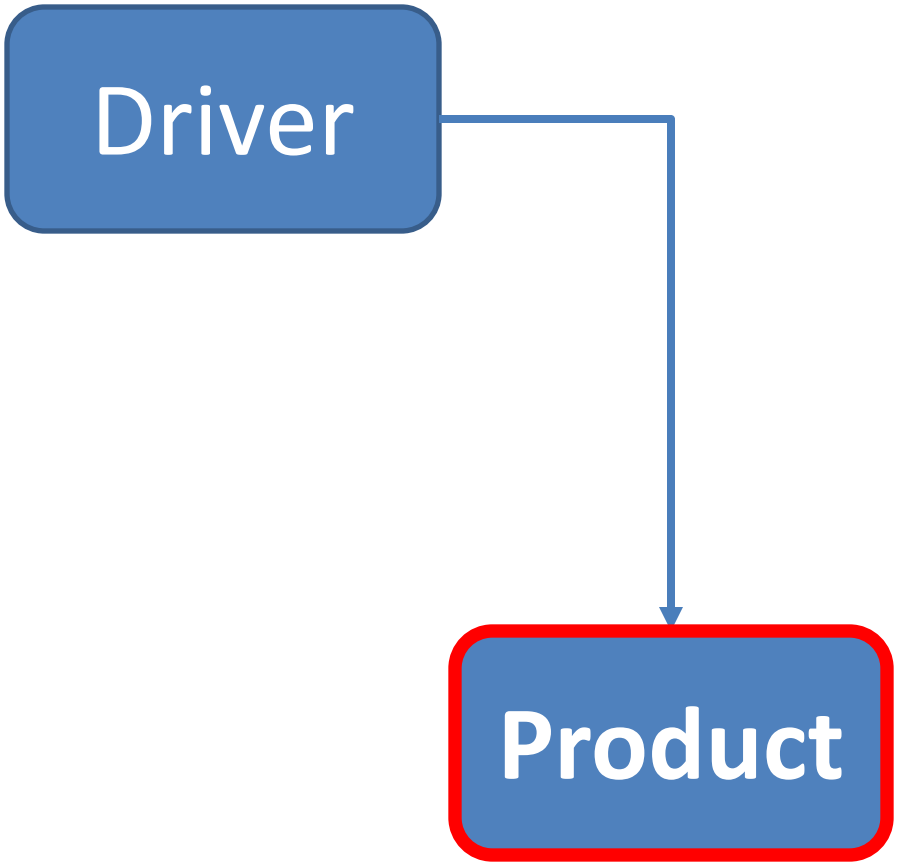
.nextInt()

.nextDouble()

.nextLine()

.next().charAt(0)

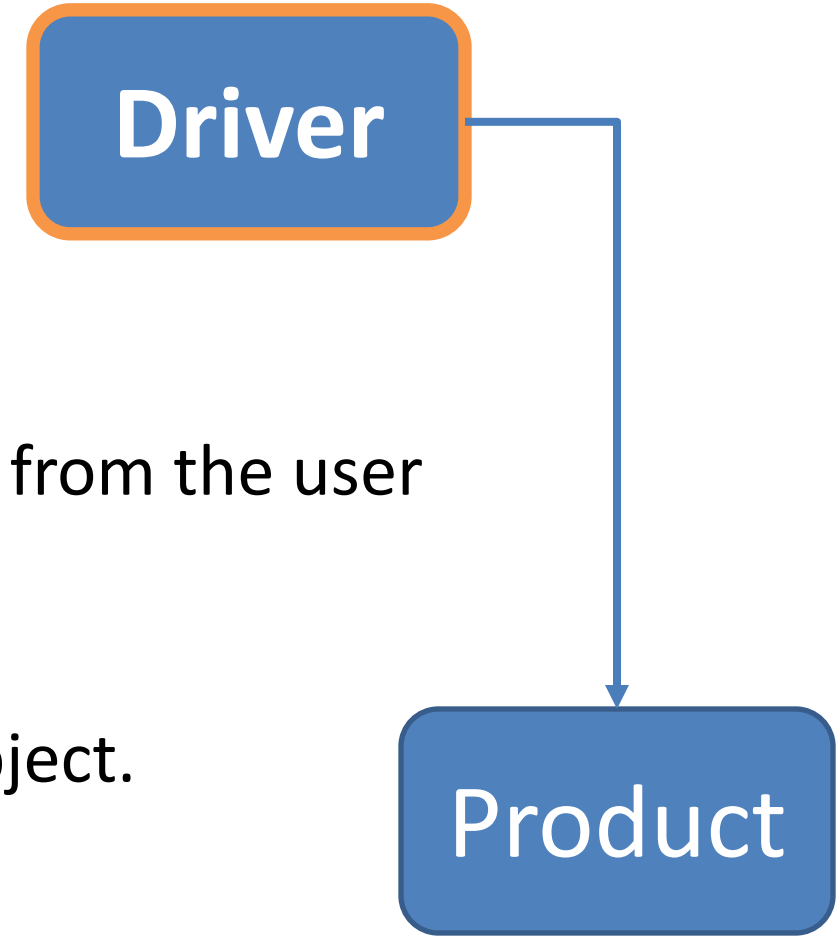
Recap: Shop V1.0 - **Product**



- The **Product** class stores **details** about a product:
  - name
  - code
  - unit cost
  - in the current product line or not?

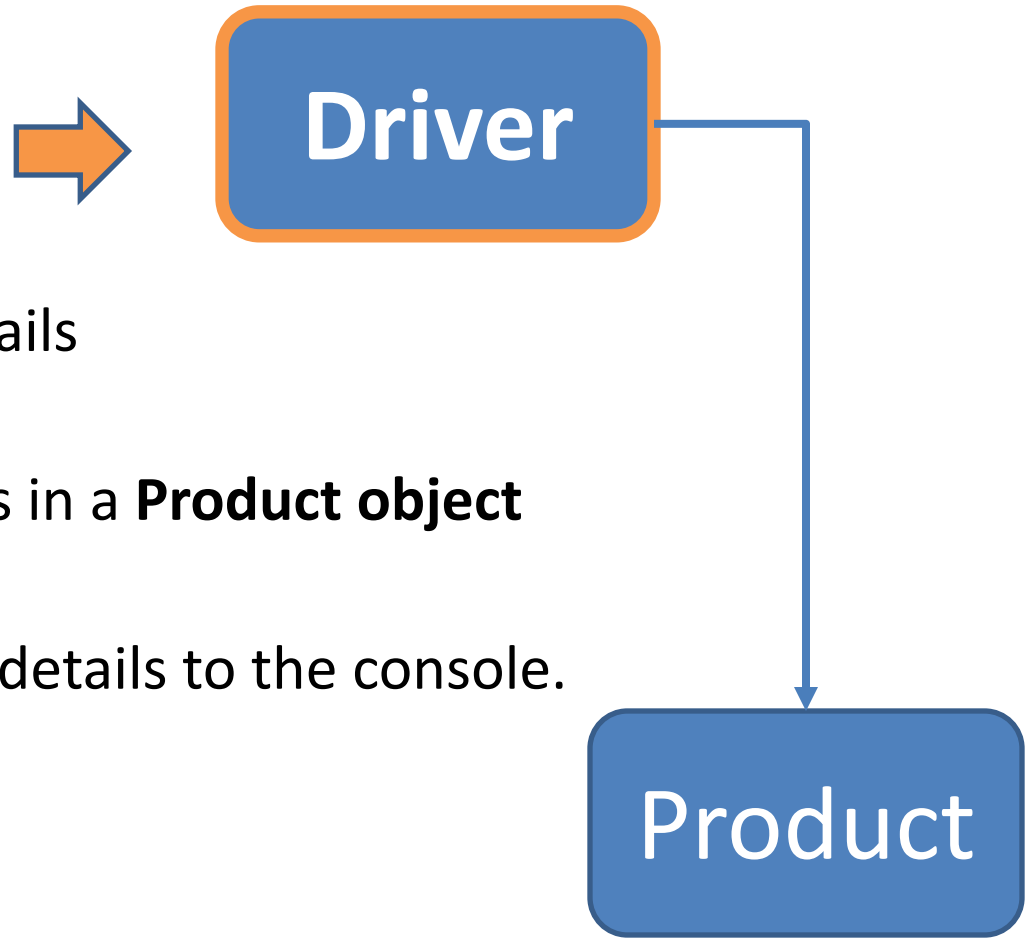
Recap: Shop V1.0 - Driver

- The Driver class
  - has the **main()** method.
  - **reads** the product details from the user (via the console)
  - **creates** a new Product object.
  - **prints** the product object (to the console)

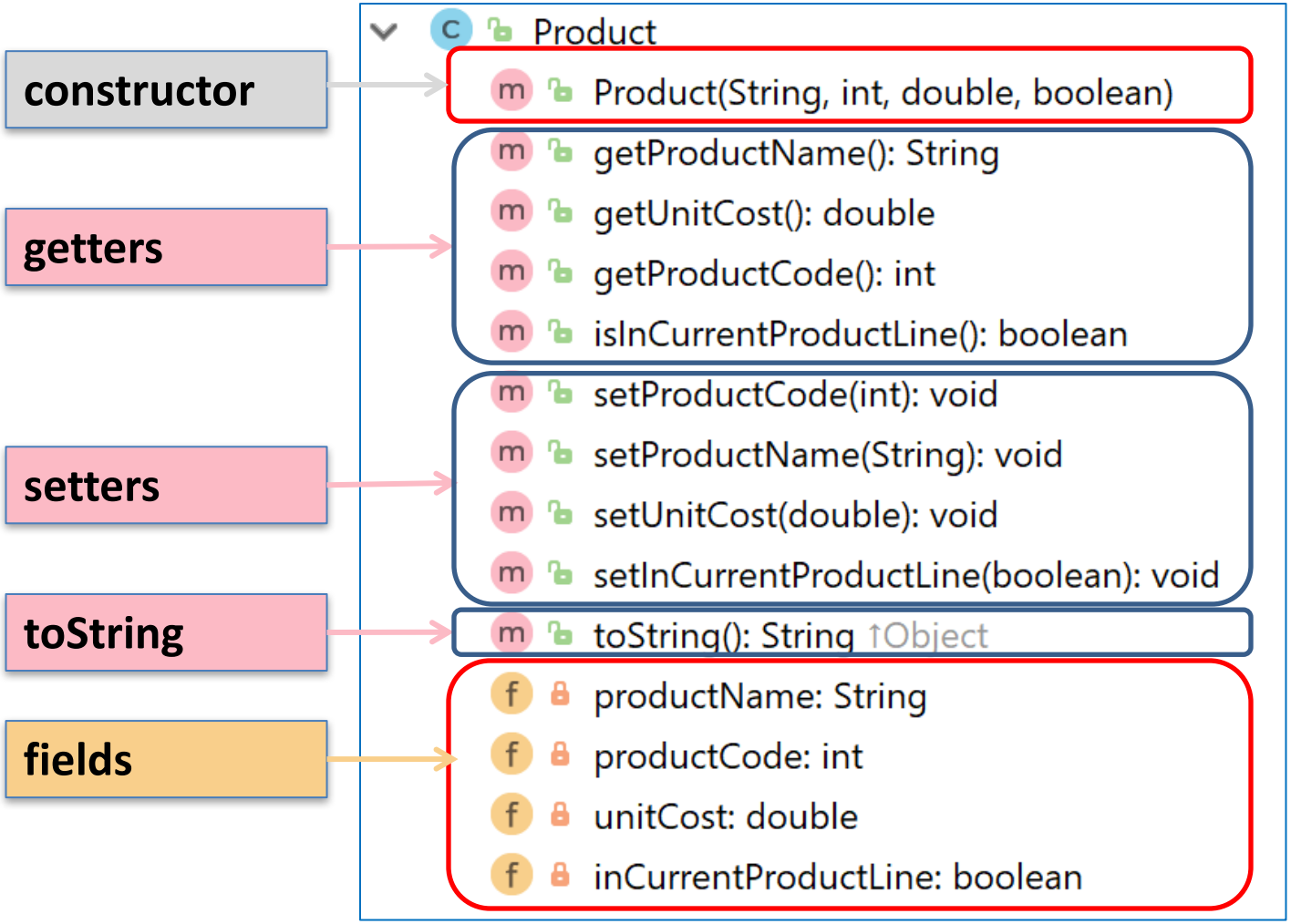


Shop V1.0 - Driver

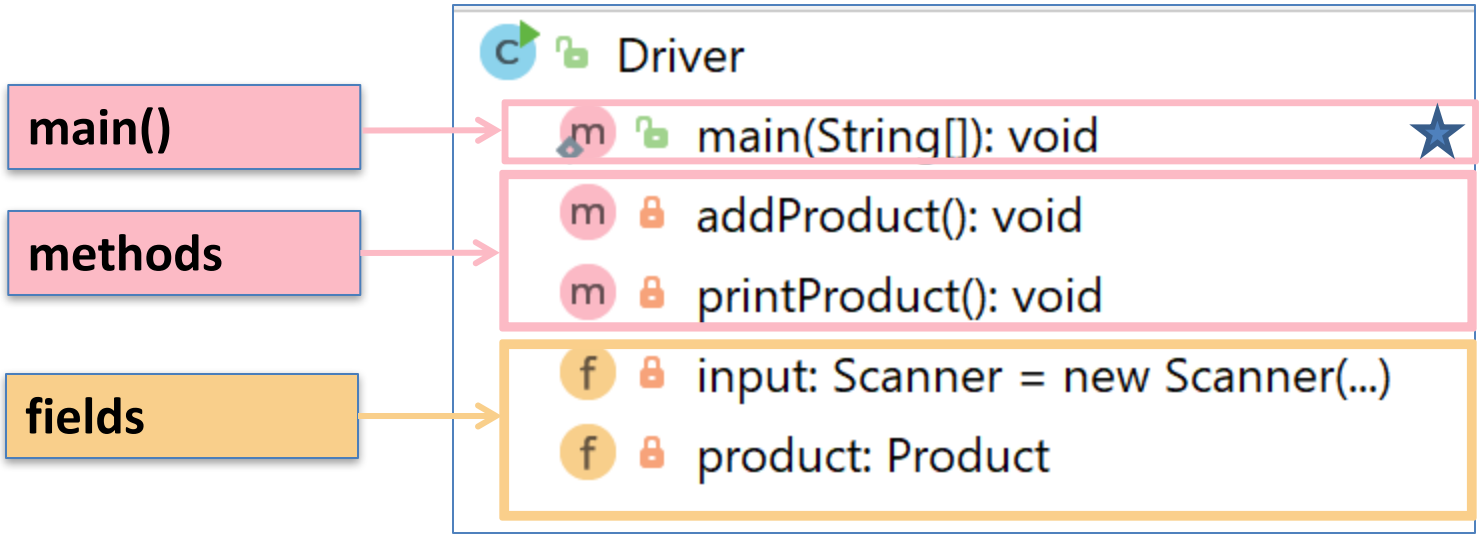
- In **Driver**,
  we want to use **Scanner**:
  - to **read in** product details
  - and **store** these details in a **Product object**
  - So we can **print** these details to the console.



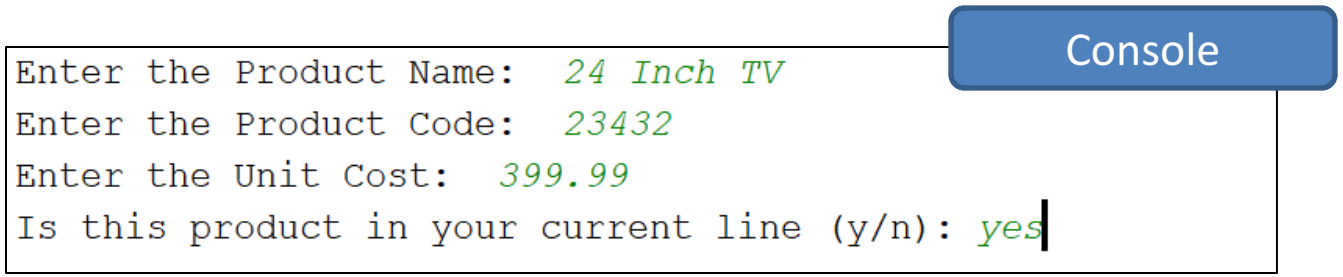
Recap: Shop V1.0: Product



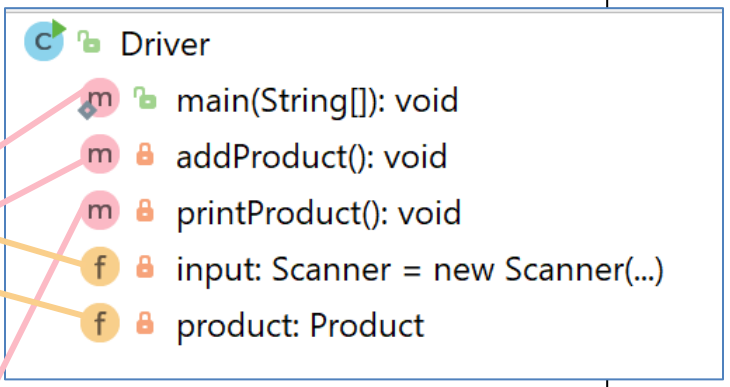
Shop V1.0: **Driver** class...



What the program looks like



```
public class Driver{

   private Scanner input = new Scanner(System.in);
   private Product product;

   public static void main(String[] args) {

      Driver c = new Driver();//Create a new Driver object c
      c.addProduct();          // Initialise c
      c.printProduct();        // Printout c
   }

   //gather the product data from the user and create a new product.
   private void addProduct(){
      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;

      product = new Product(productName, productCode, unitCost, inCurrentProductLine);
   }

   //print the product (the toString method is automatically called).
   private void printProduct(){
      System.out.println(product);
   }
}
```



Read in a **string**

Read in an **int**

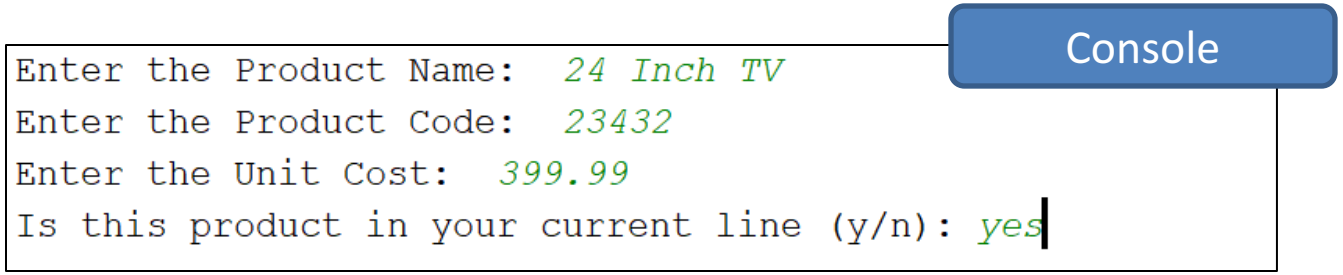
Read in an **double**

Read in an **char**

Set **boolean** based on char value

Create a new product object using the input values

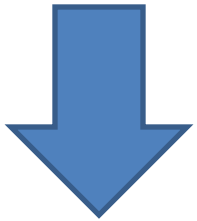
Again the **addProduct()** method does this



Now, Let's Look at how this is done…

ShopV1.0 – read Product Name (**String**)

```
System.out.print("Enter the Product Name:   ");
String productName = input.nextLine();
```

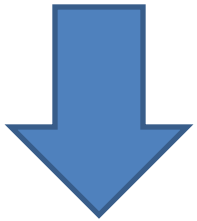


Console Output

```
Enter the Product Name:   24 Inch TV
```

ShopV1.0 – read Product Code (**int**)

```
System.out.print("Enter the Product Code:   ");
int productCode = input.nextInt();
```



Console Output

```
Enter the Product Code:   23432
```

ShopV1.0 – read Unit Cost (**double**)

```
System.out.print("Enter the Unit Cost:   ");
double unitCost = input.nextDouble();
```

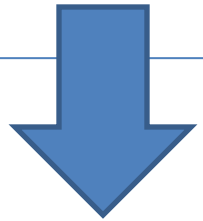
```
Enter the Unit Cost:   399.99
```

Console Output

ShopV1.0 – In Current Product Line? (**boolean**)

For **booleans**, take in a **char**acter first, then test it

```
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;
```



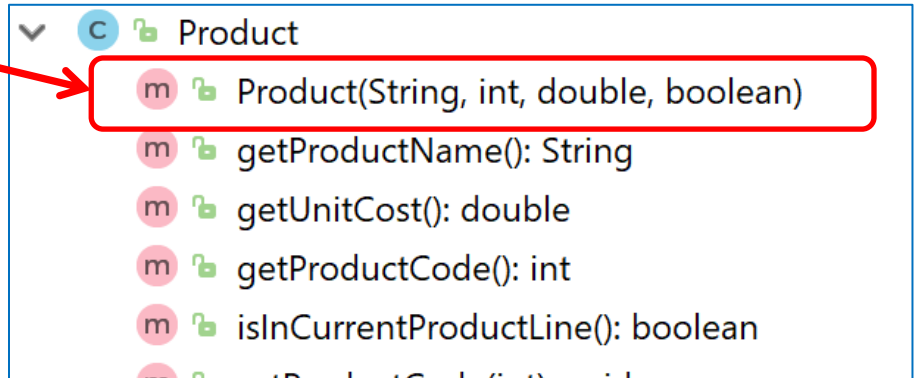
Console Output

```
Is this product in your current line (y/n): yes
```

ShopV1.0 – Create Product Object

```
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("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;

product = new Product(productName, productCode, unitCost, inCurrentProductLine);
```

Using the values taken in
pass them to the **Product constructor**



Summary

- **main()**
- **Scanner** class
  - To take in input from the console
  - First import the Class (first line)
    - `import java.util.Scanner;`
  - Then create an object variable e.g. input:
    - `Scanner input = new Scanner(System.in);`
  - Now, you can use that variable with Scanner Methods including:
    - **.nextInt()**
    - **.nextDouble()**
    - **.nextLine()**
    - **.next().charAt(0)**

Questions?