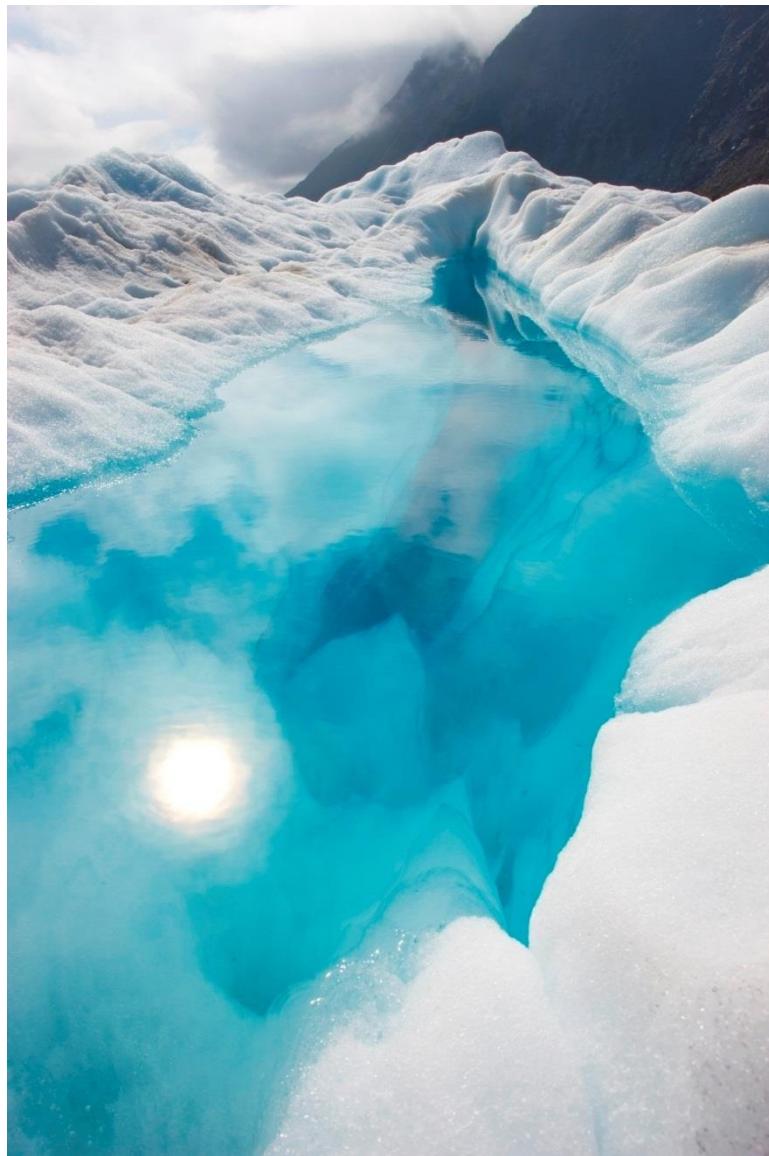


GrD5, UP811216, UP812110 and UP811828



# An Ordering System

AN ORDERING SYSTEM DESIGNED FOR LONG PIPES LTD

UP811216, UP812110 and UP811828 | ADPROC | 04/12/17

## Contents

Introduction .....	1
UML and Instance diagrams (Object).....	7
UML Case Diagram .....	13
Testing schedule .....	14
Testing.....	15
Sample usage.....	84
Source code .....	94

## Introduction

We have been given the opportunity to design and create an application for Long Pipes ltd. This program is intended to allow users to receive quotes for different prices for the different types of pipes that Long Pipes supply.

Our application must determine if the type of pipe, defined by the options that the user set out, can indeed be supplied by Long Pipes, if this is the case, then Long Pipes will give the user a quote on this particular Pipe that they require in this order.

The user chooses the following options to our application for their requested pipe:

1. Plastic grade value (From 1 to 5)
2. The color print (This is the number of colors on the pipe, this can be no color, 1 color or 2 colors)
3. Inner insulation – Choice between yes and no
4. Outer reinforcement – Choice between yes and no
5. Chemical resistance -Choice between yes and no
6. Length (From 0.01 m to 6m)
7. Outer diameter – (From 0.01 inches to 2 inches)
8. Quantity – (From 1 to 100)

Upon the user entering these values and options, our system will determine if the pipe can be supplied, if this is the case, then this pipe is added onto the user's order,

otherwise we provide an error message to tell the user that Long pipes does not supply the pipe with these requested options.

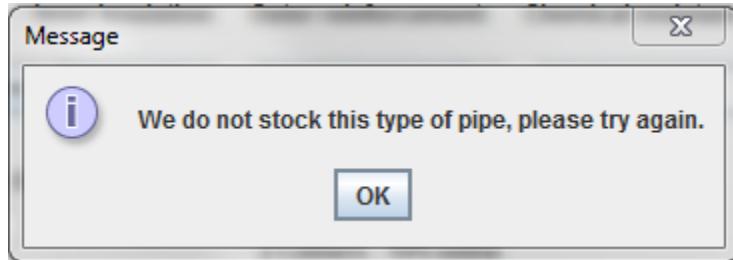


Figure 1 – Error message from selected pipe options

The user is allowed to at any time during their time with the application to use these options to optimize their order from Long Pipes:

1. Remove an order – Delete the whole order and start again or continue with the other orders
2. Remove a pipe – Delete the requested pipe from the current order
3. Create multiple orders, the user can create as many orders as they please and each order subtotal is calculated and shown
4. Quote different pipes in the same order – A user is allowed to add as many different pipes, as long as Long Pipes can supply them to a single order and edit as they go.
5. Receive a receipt for their order(s) at the end of their ordering session, by which all products for all order at their order subtotal will be shown as 1 total cost and a list of products from all orders.

After the user is happy with the products they have selected for all orders, the user can then receive a receipt for their products. In this receipt, the user will be shown the total cost for all orders and the products that the user has selected for their orders.

#### Assumptions:

- It would be out of place to include a range for the outer diameter over 5 Inches.
- It is important to minimize the risk of user error to a minimum therefore we have only used 3 input fields.
- We wanted to allow users to make as many orders as they wish and have as many products as they see fit in each order, our design reflects this.

- We assume that users will create multiple orders in same session of use as a way of "organizing" pipes.
- That a user will wish to see a receipt and this should be a separate screen.
- We assume users want a guide.
- Product information will only display relevant information about the product.
- We assume users want to see the sub-total and total of all orders.
- Users may wish to remove all orders or a specific order
- Users may wish to remove all products or a specific product

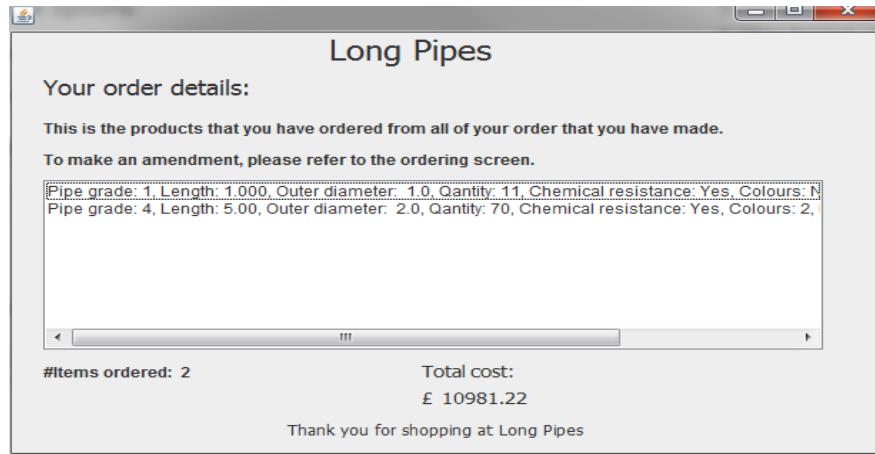


Figure 1.1 - Long pipe receipt screen

At this stage still, the user is allowed to change their orders, they can go back to their orders and remove products as they please as well as adding new products, by which the subtotal and total cost will update accordingly.

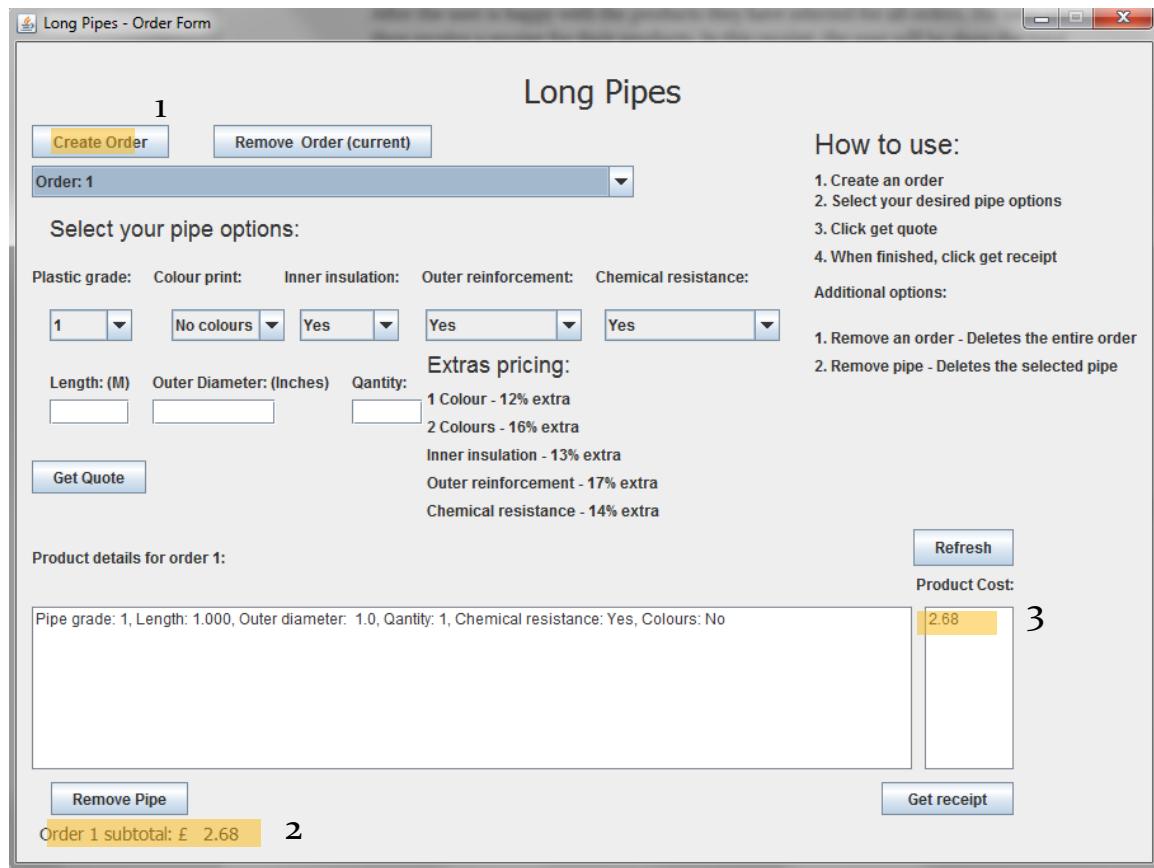


Figure 1.3 – Order 1 screen

1. The currently selected order.
2. The order subtotal cost.
3. The cost of an individual product

**Long Pipes**

**Create Order**   **Remove Order (current)**

**Order: 2**

**1 Select your pipe options:**

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1   No colours   Yes   Yes   Yes

Length: (M) Outer Diameter: (Inches) Quantity:

Extras pricing:

1 Colour - 12% extra  
2 Colours - 16% extra  
Inner insulation - 13% extra  
Outer reinforcement - 17% extra  
Chemical resistance - 14% extra

**Get Quote**

**Product details for order 2:**

Pipe grade: 5, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: Yes, Colours: 2, Insulated: Yes, Reinforced: Y

**Refresh**

**Product Cost:** 9.12

**Remove Pipe**

**Get receipt**

**Order 2 subtotal: £ 9.12** **2**

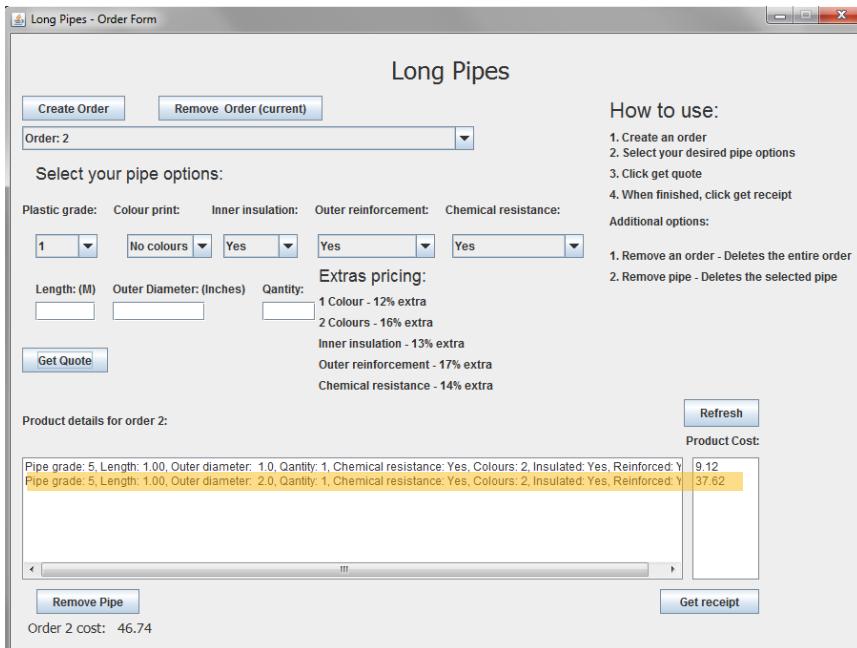
**3**

Figure 1.4-

Order 2 screen

Here we can see that for the second order (Order 2), the subtotal is £9.12 which contrasts with the previous order subtotal which was £2.68.

When we add a product to the second order, subtotal updates.



The new order subtotal for the second order is now £46.74. The receipt also shows this different total value for all order.

Before addition of new pipe:

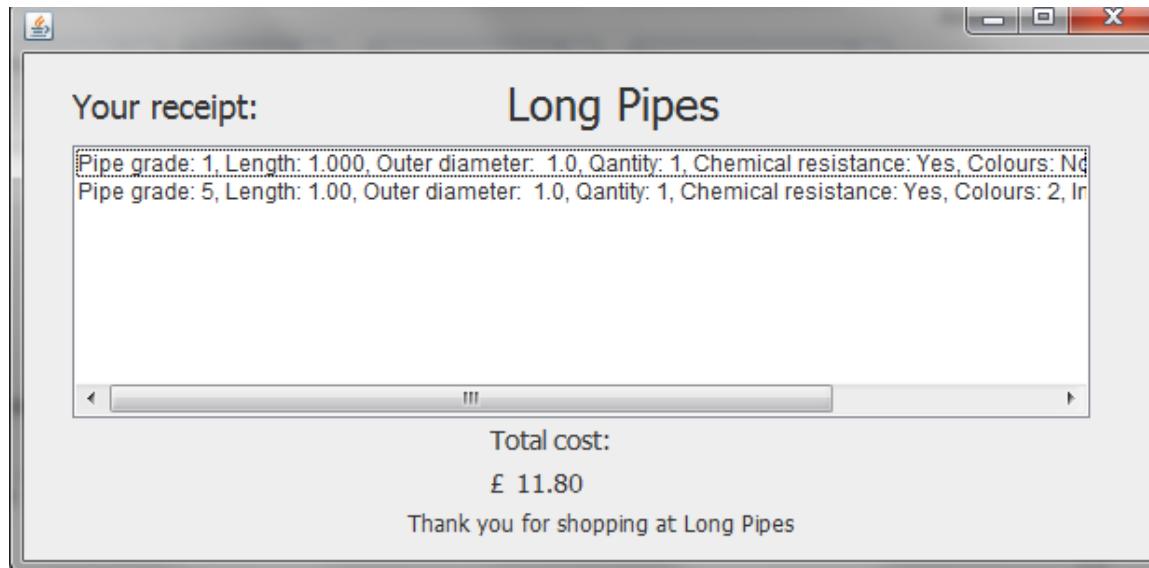
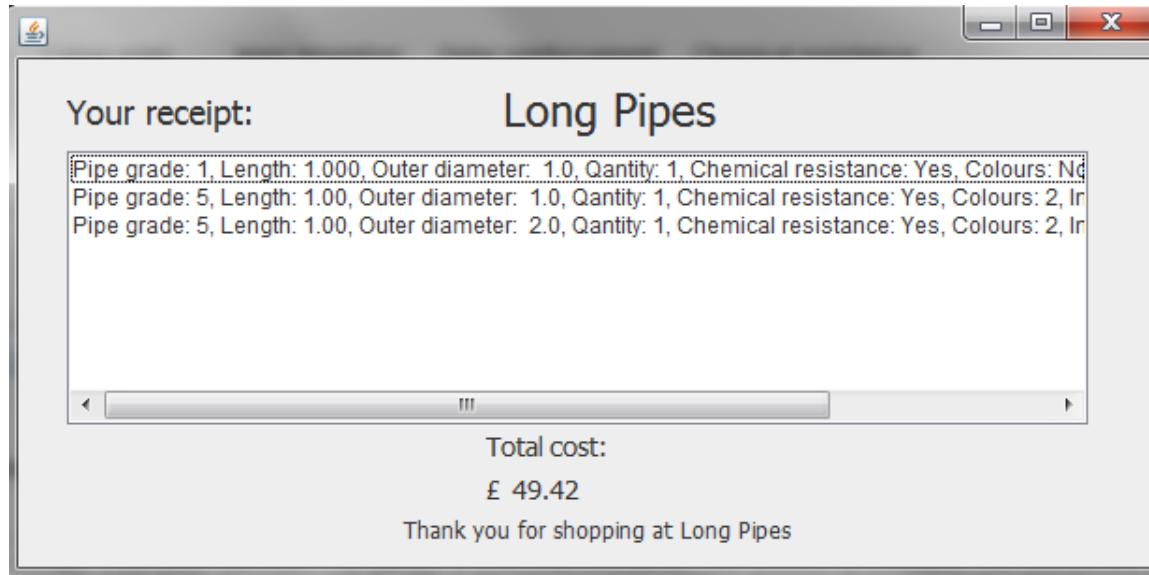


Figure 1.5 – Long pipes receipt screen before addition of a new pipe to an order

After addition of new pipe:



The new total cost is now £49.42 which reflects the addition of this new pipe, the new total cost is shown along with the new pipe in the list of products from all orders.

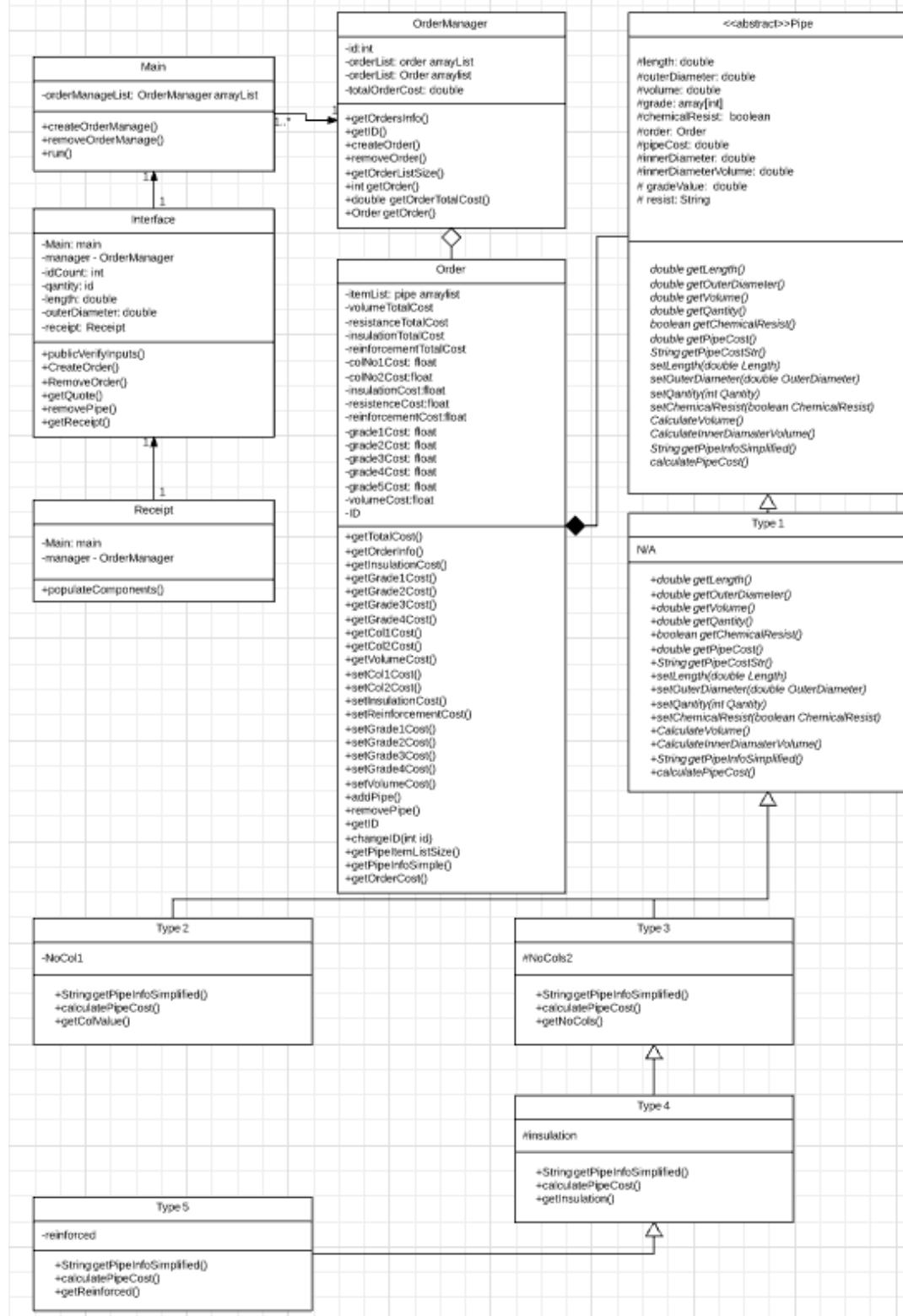
This is simple introduction to the use of the program, refer to the sample usage for fuller report of this applications' functionality.

## UML and Instance diagrams (Object)

Here we provide our UML and instance diagrams. Our UML diagram provides a different perspective of our system as opposed to the sampled introductory functionality screenshot. We show 2 UML diagrams and 1 instance diagram, our UML diagram consist of:

- . An overall view of all of the classes
- . A more detailed view of the Pipe hierarchy, detailing the various different Pipes and their inherited methods and variables.

Overall view of system:



### Description of overall system: (Detailed version)

Our program starts with our main class creates an order manager object and an interface object and pass the order manager, and the main object to the interface as parameters. Once a user clicks on create order, a new order is created and added to order manager.

From here the user is free to select their pipe options, discussed earlier, and then they click 'Get quote' in order to check if this pipe can be supplied by Long Pipes. If this is the case, then this product is now added to the list of products/pipes that the order object generates upon creation, at first it has size of 0 as there are no pipes in the product list.

When get quote is running, it executes 2 main tasks:

- Verify Inputs
- Attempt to add a pipe

#### Verify inputs:

When this method is run, the length, quantity and outer diameter values are checked. If there is an error here, either type of range, this message is displayed to the user.

Otherwise the method return a true value task 2 proceed to run.

#### Attempt to add Pipe

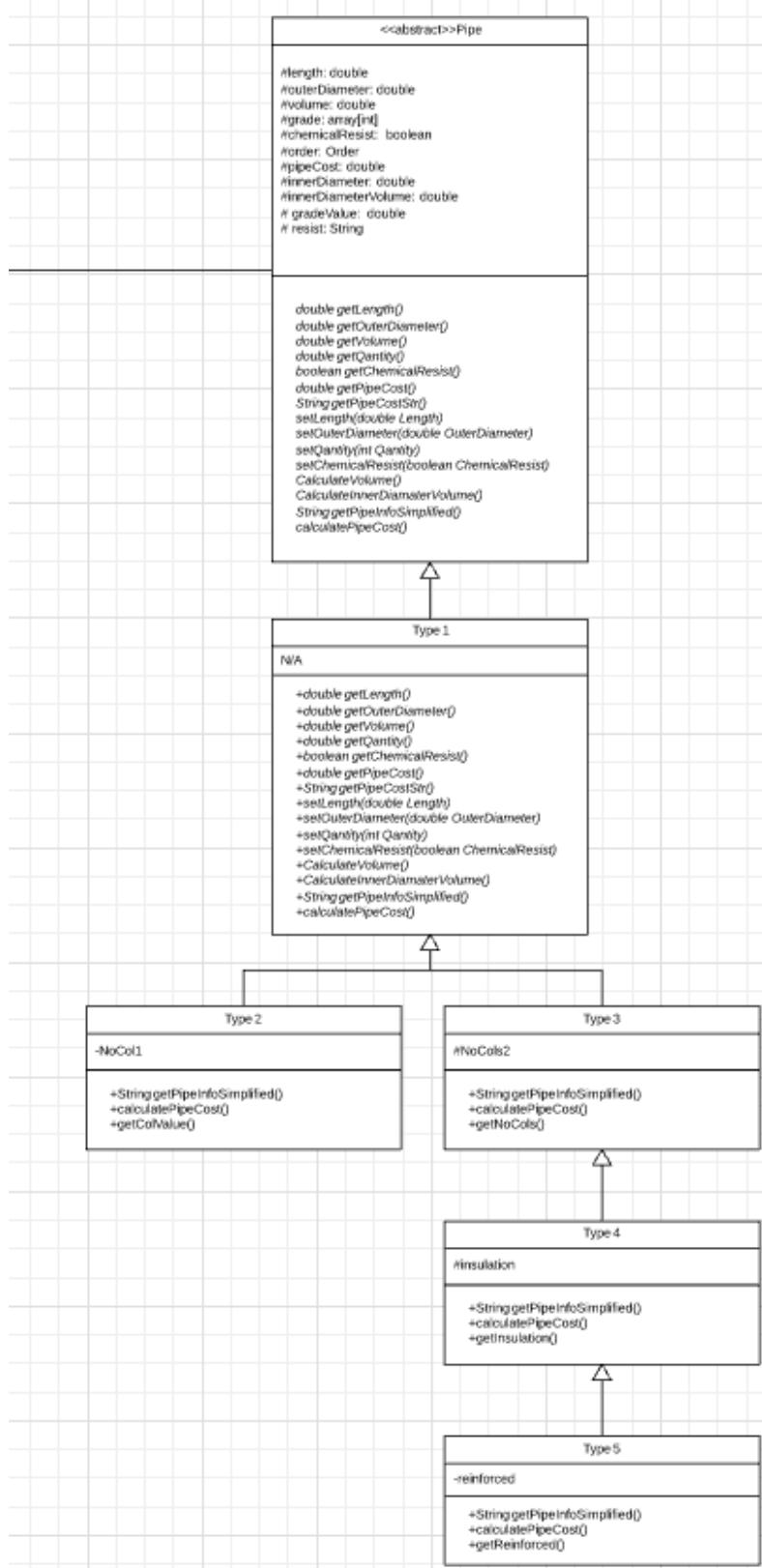
In this task, the Get Quote method will now try and match the user's information with a Pipe that is supplied. If there is a match i.e. a user's options correspond with a specific type of pipe, a message is displayed to the user. This new pipe is added to the user's selected order and the new product is shown on-screen as well as supporting information such as product cost. This affects the overall subtotal of the order.

After the order has been added, the user is now free to request a receipt using the 'Get Receipt' button or can remove this pipe or any other pipe previously added or create a new order or entirely remove this order. The program will respond accordingly.

If the user clicks 'Get Quote' but wish to amend their order(s), the ordering screen is still displayed and thus the user can still change an order as they desire. This could be, adding a new pipe, removing an order etc.

When the user has finished using the program, after they have clicked 'Get Receipt', they are free to view the contents of all of their orders including the total cost and number of items in all orders combined.

Detailed view of Pipe Hierarchy:



Description of pipe hierarchy:

Pipe is an abstract class whereby only the methods without any implementations are here, our types of pipes inherit these methods and implement them in their own specific way. Our constructor for this class uses a number of protected variables that are used by the inherited classes.

View the inheritance above before continuing.

In our ordering system, type 1 pipe does not include any extra details associated with it, it only contains calculations for the grade and chemical resistance values. Therefore, it is acceptable to include only a first set of implementation of the methods that you see in the Pipe abstract class. The only exceptions here are, getPipeInfoSimplified(), calculatePipeCost() methods which are specifically implemented differently in the respective classes below type 1.

All classes above type 1, 2-5 all inherit from type 1 in their respective characteristics leading us to our Pipe class hierarchy.

To provide a simple snapshot of what functionality is exiting, view below:

. Type 5 inherits methods and variable from type 4, type 3 and type 1. This allows for polymorphism as type 5 will use its own implementation for getPipeInfoSimplified() instead of either type 4, 3, or 1 implementation as type 5 is inherited as a child of the previous classes.

All of our methods which are used in their respective classes are shown as well as their visibility:

- Private
- Protected
- Public.

Instance diagram:

Here we have chosen to show an instance diagram for when a pipe of type 5 is added to their order.

t5: Type 5
length = 5 outerDiameter = 2 gradeValue = 5 chemicalResist = true order = Order1 resist = "Yes" quantity = 1 insulated = true reinforced = true colours = 2 volume = 5.87 pipeCost = 9.29

Here we can see that when a new pipe of this type is made, parameter such as (length, outerDiameter, gradeValue, chemical Resist and quantity) are given and their values:

- Length = 5
- OuterDiameter = 2
- GradeValue = 5
- ChemicalResist = true
- Quantity = 1

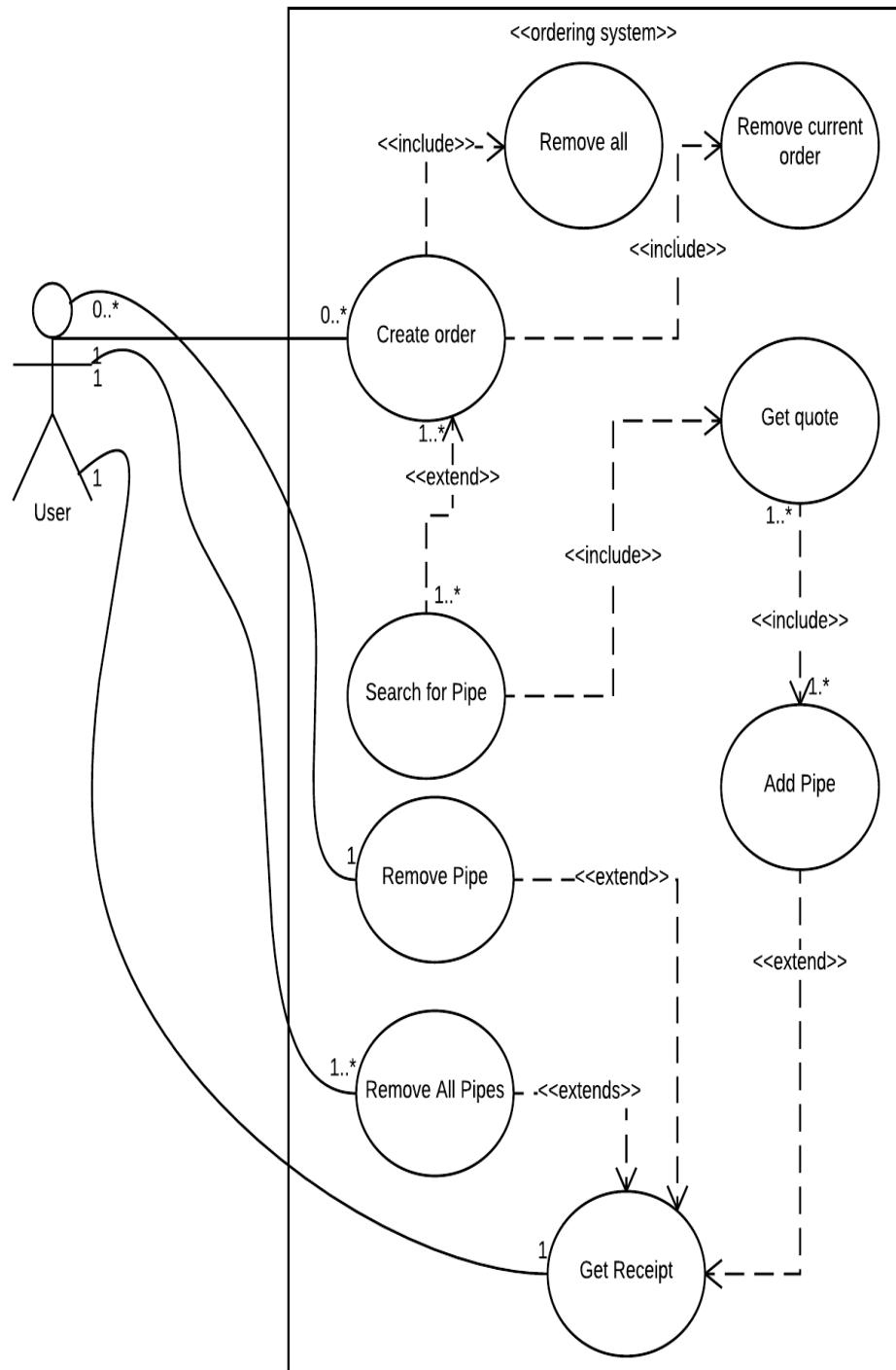
Are supplied, this means that the pipe volume is 5.87 (Inches) and the pipe cost is £9.29.

The default values for this pipe are automatically set:

- Order – The order that this pipe is a part of.
- Insulated - The default value for this type of pipe which is always true.
- Colours – The default value for this type of pipe is 2.
- Reinforced – The default value for this type of pipe is true.

These length; outerDiameter; GradeValue; ChemicalResist and Quantity are sample supplied inputs and can changed and will affect the overall cost of this type of pipe.

## UML Case Diagram



The UML case diagram shows all of the functions/tasks that the user can do within the ordering system.

The 2 main tasks that the user can do are:

- Creating multiple order
- Getting a receipt

After the user has created at least 1 order and has at least 1 pipe/product within this order, they can then display the receipt screen.

After creating an order, the user has now access to:

- Search for a pipe
- Getting a quote – This checks to see if the inputs are acceptable and then checks to see if the pipe can be added. If this is the case, the pipe is added, otherwise an appropriate error message is displayed to the user.
- Remove all or the currently selected pipe from the order - The user can now delete a selected pipe or delete the entire order.

#### Get Receipt task:

The user can receipt a receipt for all of their orders and the products contained within them. The total cost as well as all of the products are now displayed form all orders. At this point, the user can now close the program or amend an order.

#### Order:

Removing an order or all of them - The user is free to delete a selected order or they can now remove all orders that have been made. The program responds accordingly.

## Testing schedule

ID:	Area of interest:	Reason:
1	Initial run-time status including: . Display of appropriate labels . Display of appropriate sections Button functionality including: . Remove order . Remove all orders . Get Quote . Remove Pipe	In order to check that all of the correct labels are present and show the right information.  To ensure that the buttons are responding correctly at this point in the program.

	. Get receipt	
2	In use – Create order and remove functionality  Button functionality including: . Get Quote . Remove Pipe . Get receipt	To check that creating and removing orders works as expected.  To ensure that the buttons are working with no products added to any orders.
3	In use, adding a pipe Button functionality: . Get Quote . Get receipt	To ensure that the right information is shown after successfully adding a pipe.  To ensure that button functionality works as expected.
4	In use, products have been added: Button functionality: . Remove Pipe . Remove all pipes	To ensure that button functionality works as expected.
5	In use, multiple order selection	To ensure that information for multiple orders is being displayed with appropriately.
6	Calculation of an order	To ensure that the calculation of an order is correct.
7	Calculation and update of multiple orders	To ensure that calculation of multiple orders is correct.
8	Receipt screen information	To ensure that the receipt screen is displaying the right information about all orders.

## Testing

Area 1 – Initial run-time status

In this area we are looking for the following results:

1. Display of appropriate labels
2. Display of appropriate sections
3. Remove order functionality – Appropriate error message (On click)
4. Remove all orders functionality – Appropriate error message (On click)
5. Get Quote functionality – Appropriate error message (On click)

6. Remove Pipe functionality – Appropriate error message (On click)
7. Remove All Pipes functionality – Appropriate error message (On click)
8. Get receipt functionality – Appropriate error message (On click)

Test 1 – Display of appropriate labels:

When the program loads, initially we should expect to see no subtotal calculated, no products displayed and no orders made. Hints should be provided on-screen as to how to use the program.

Initial run-time:

1. 1 and 4 - No product details can be shown at the moment as no orders have been made at the moment, the text here provides us an insight on what the user needs to do to start using the program.
2. 2 - This section is useful and displayed appropriately because it clearly tells the user how to use this application from the initial run-time state.
3. No cost can be calculated at the moment as no order have been made, therefore this is correctly shown.

This test has passed.

### Test 2 – Display of appropriate sections

Long Pipes

Create Order   Remove all orders   Remove Order (current)

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1   No colours   Yes   Yes   Yes

Length: (M) Outer Diameter: (Inches) Quantity: 2 Colours - 16% extra  
Inner insulation - 13% extra  
Outer reinforcement - 17% extra  
Chemical resistance - 14% extra

Get Quote   Refresh   Product Cost:

Get receipt

How to use:  
 1. Create an order  
 2. Select your desired pipe options  
 3. Click get quote  
 4. When finished, click get receipt  
 Additional options:  
 1. Remove an order - Deletes the entire order  
 2. Remove pipe - Deletes the selected pipe

1 Colour - 12% extra  
2 Colours - 16% extra  
Inner insulation - 13% extra  
Outer reinforcement - 17% extra  
Chemical resistance - 14% extra

Unable to show product details for an order at the moment, consider making an order.

Remove Pipe   Remove all pipes   Unable to calculate order subtotal. 3

In the diagram above, we can see that the extras pricing is shown, along with the information regarding using the application is apparent. The receipt screen is not showing which is expected here.

Other sections such as:

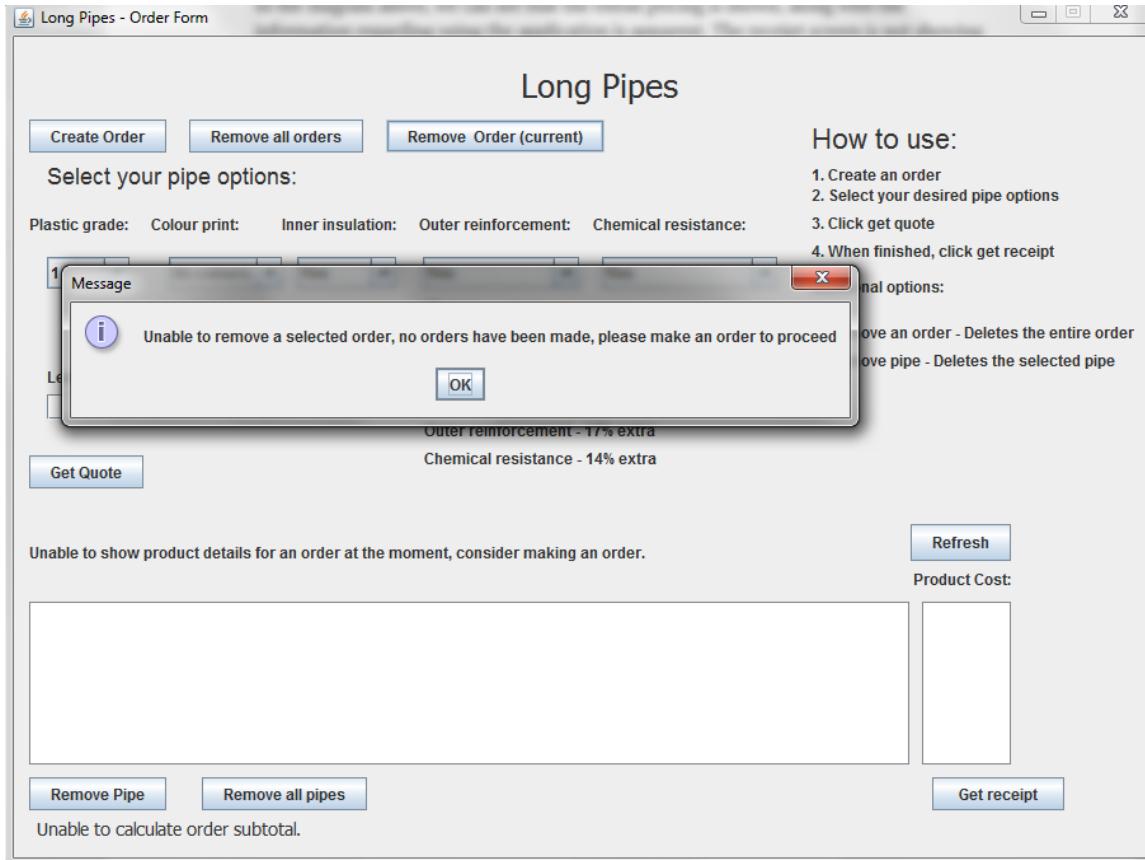
- Select pipe from order and its input fields
- Get quote is positioned correctly
- Remove pipe/all pipes is positioned correctly
- Get Receipt is positioned correctly.
- Remove order/all order is position correctly.

This test has passed.

### Test 3 - Remove order functionality

In this test we are looking for an appropriate error message when a user attempts to remove an order. We expect to see an error message detailing that an order cannot be removed as no order have been placed at the moment.

On click response of remove order:



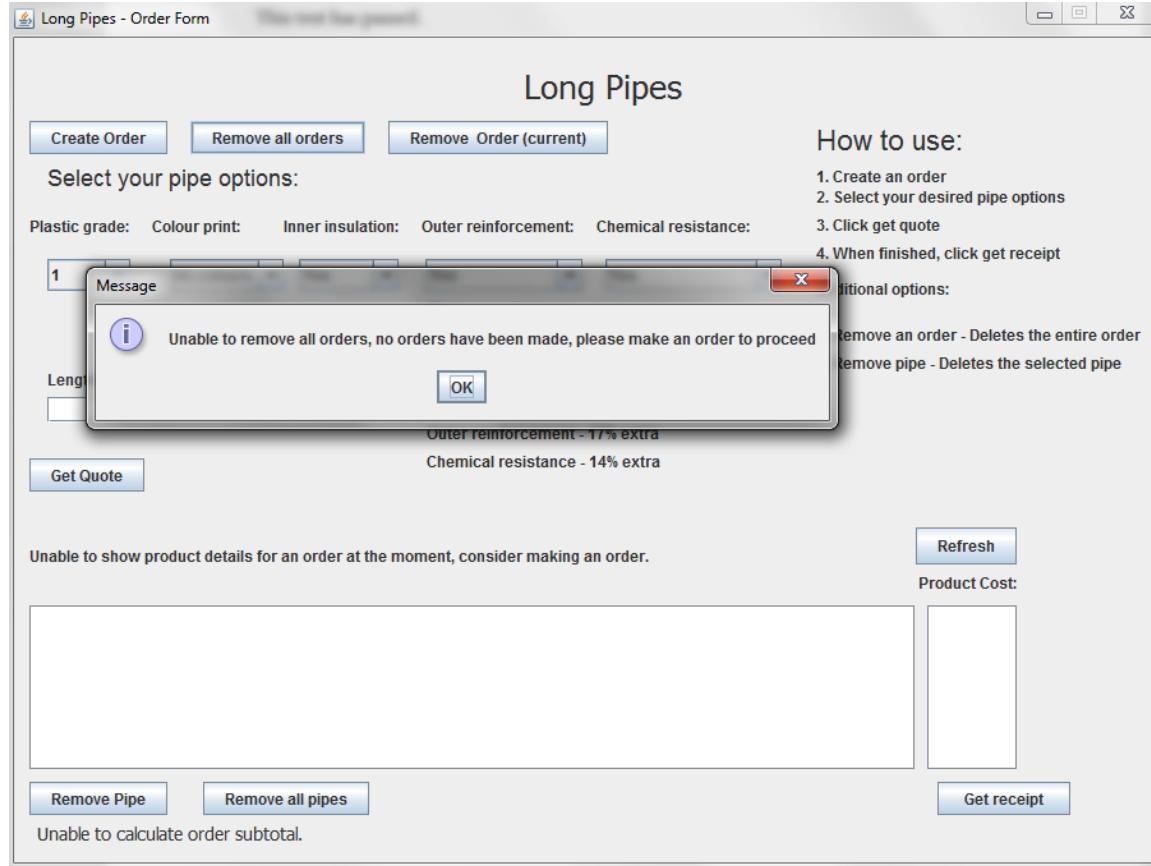
We can see here that this button is working as expected as the error message is displayed for this type of user error.

This test has passed.

#### Test 4 – Remove all order functionality

In this test we are looking for an appropriate error message when a user attempts to remove all orders. We expect to see an error message detailing that an order cannot be removed as no orders have been placed at the moment.

On click response of remove all orders:



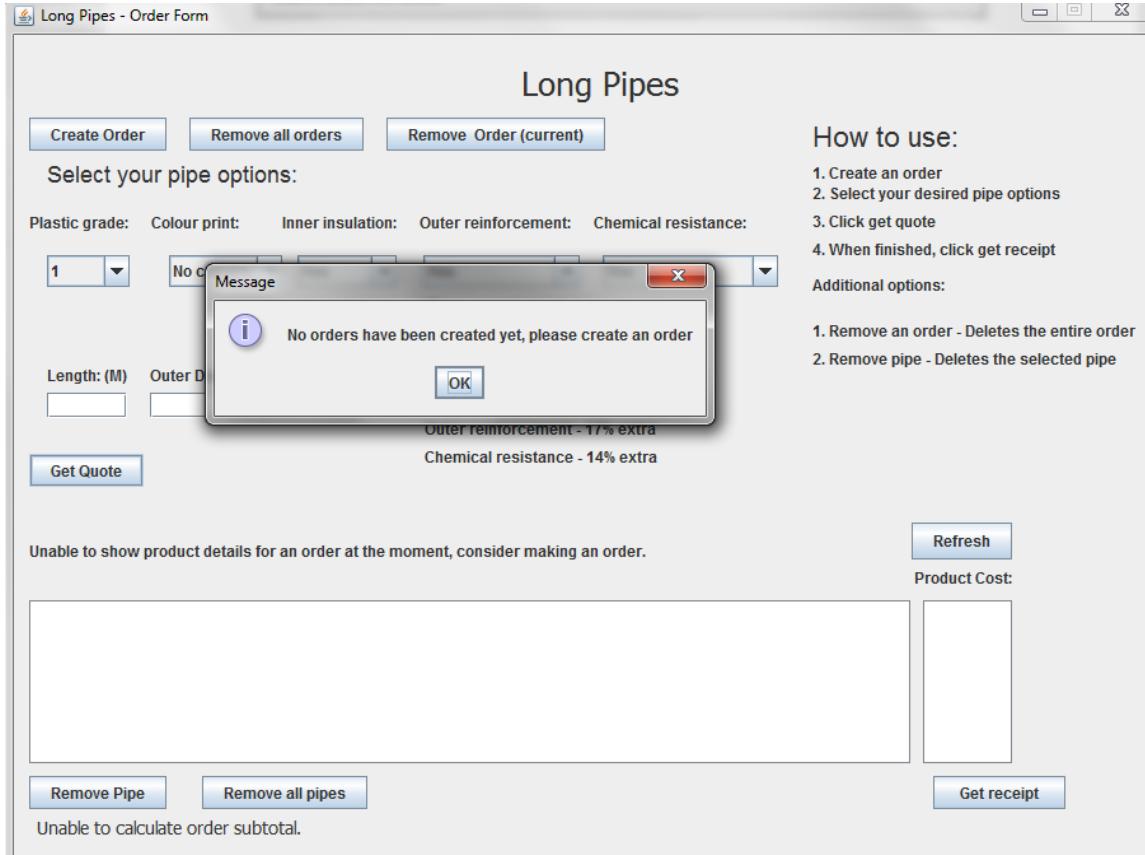
We can see here that this button is working as expected as the error message is displayed for this type of user error.

This test has passed.

#### Test 5 - Get quote functionality

In this test, we expect to see an error message ad no quote i.e. price for a product can be given at the moment as order have been made.

On click response of Get quote:



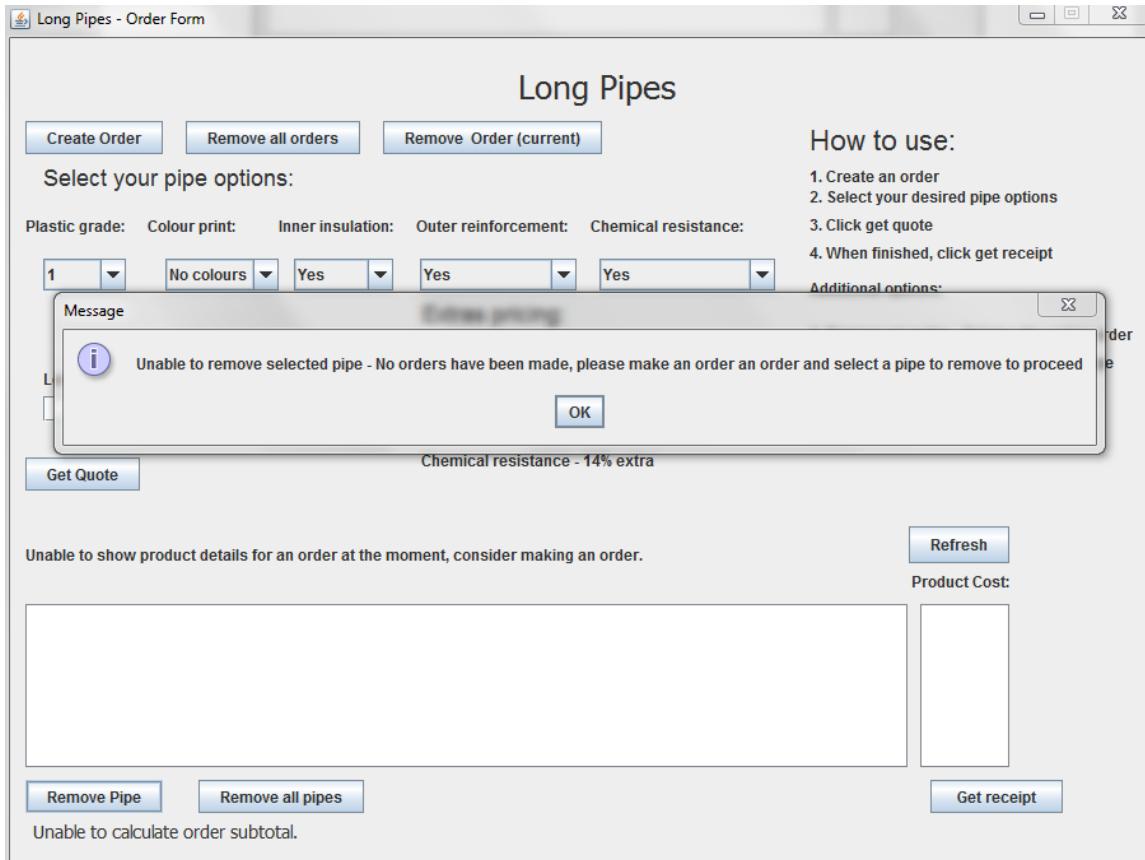
We can see here that this button is working as expected as the error message is displayed for this type of user error.

This test has passed.

### Test 6 – Remove pipe functionality

The user should receive an error message here as the program is unable to remove a pipe when even a single order has not been made.

On click response of remove pipe:



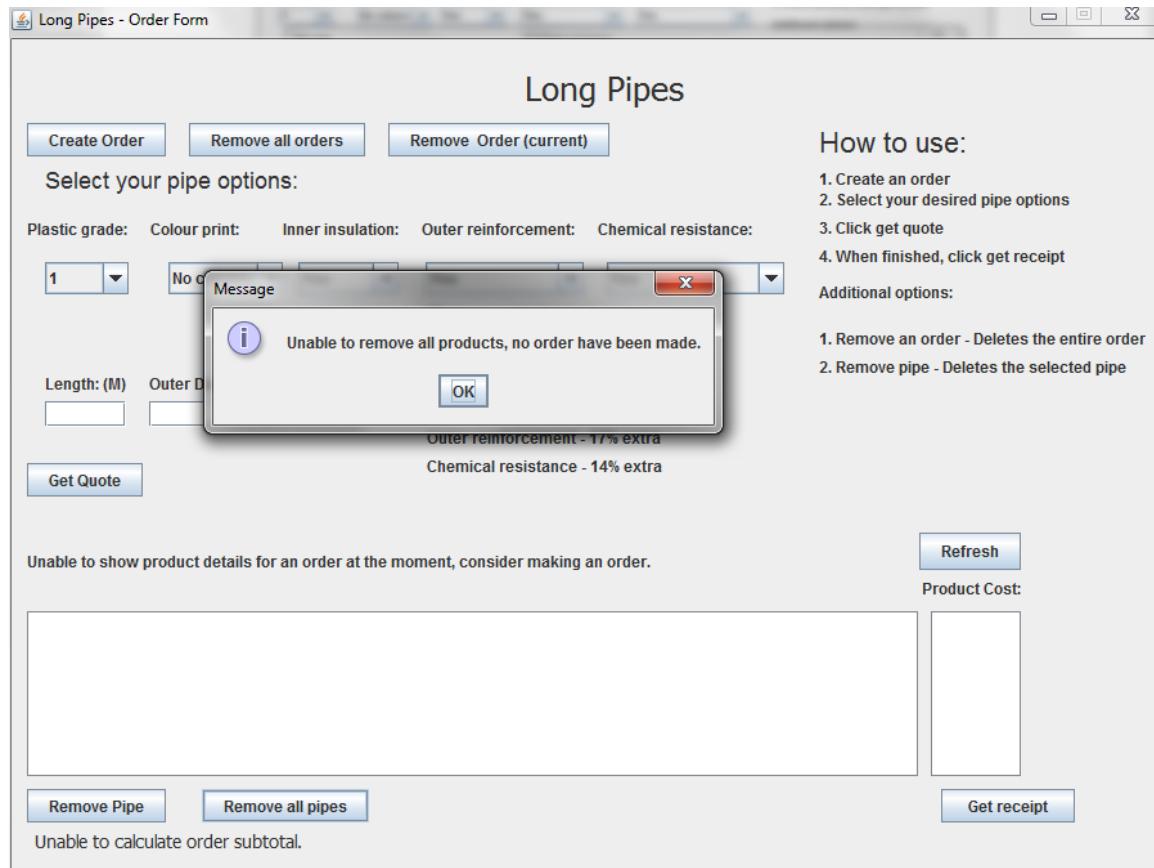
We can see here that this button is working as expected as the error message is displayed for this type of user error.

This test has passed.

### Test 7 – Remove all pipes

Again like the previous example, we cannot remove all pipes from an order that hasn't been made.

On click response of remove all pipes

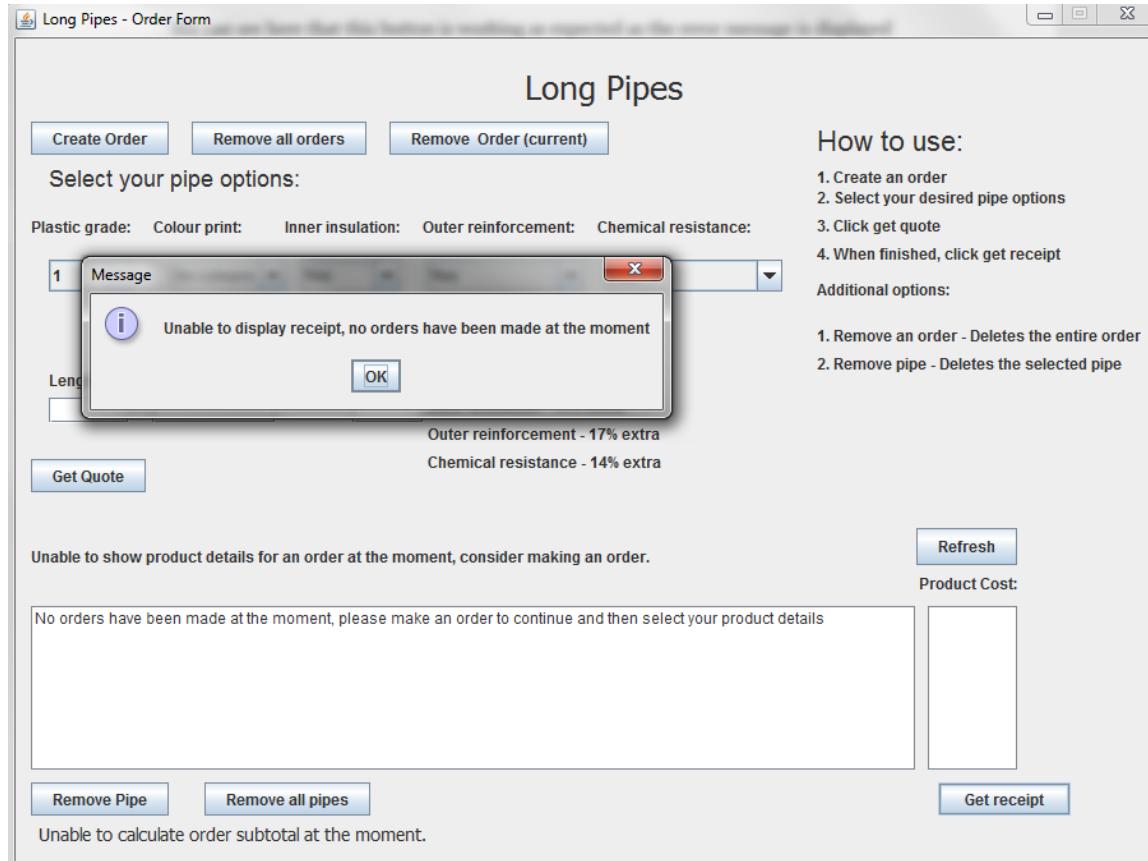


We can see here that this button is working as expected as the error message is displayed for this type of user error.

This test has passed.

### Test 8 – Get receipt functionality

If no orders have been made, then an error message should show to the user, explaining that as no orders have been made, the program is unable to show a receipt at the moment.



We can see here that this button is working as expected as the error message is displayed for this type of user error.

This test has passed.

### Area of Interest 1 – Initial run-time status:

1. Display of appropriate labels - PASSED
2. Display of appropriate sections - PASSED
3. Remove order functionality – Appropriate error message (On click) - PASSED
4. Remove all orders functionality – Appropriate error message (On click) - PASSED
5. Get Quote functionality – Appropriate error message (On click) - PASSED
6. Remove Pipe functionality – Appropriate error message (On click) - PASSED
7. Remove All Pipes functionality – Appropriate error message (On click) - PASSED
8. Get receipt functionality – Appropriate error message (On click) - PASSED

Area summary:

8/8 PASSES – This section has passed.

Area of interest 2 – Create order and remove order button functionality:

In this area we are looking for the following results after an order has been made:

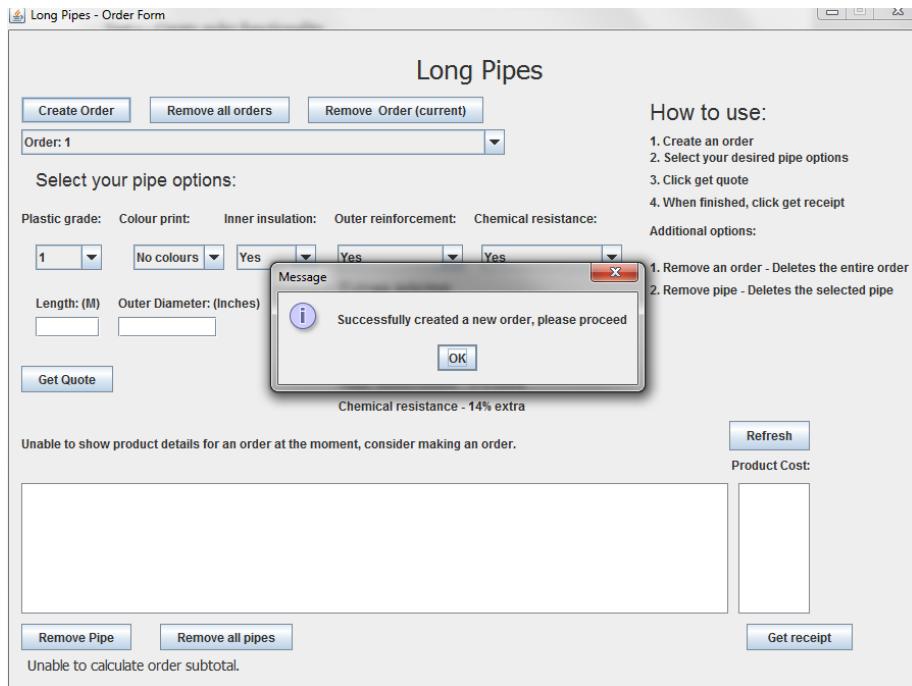
1. Create order button makes a new order and this is shown
2. Remove order button deletes this order
3. Remove all order button removes multiple orders
4. Get Quote functionality – Appropriate error message (On click)
5. Remove Pipe functionality – Appropriate error message (On click)
6. Remove All Pipes functionality – Appropriate error message (On click)
7. Get receipt functionality – Appropriate error message (On click)

Test 1 – Create order functionality

When a user clicks create order, a message should appear stating that an order has been successfully created, in the drop-down list below we should then see “Order: 1” present. Next we should see that the appropriate; labels have now been changed.

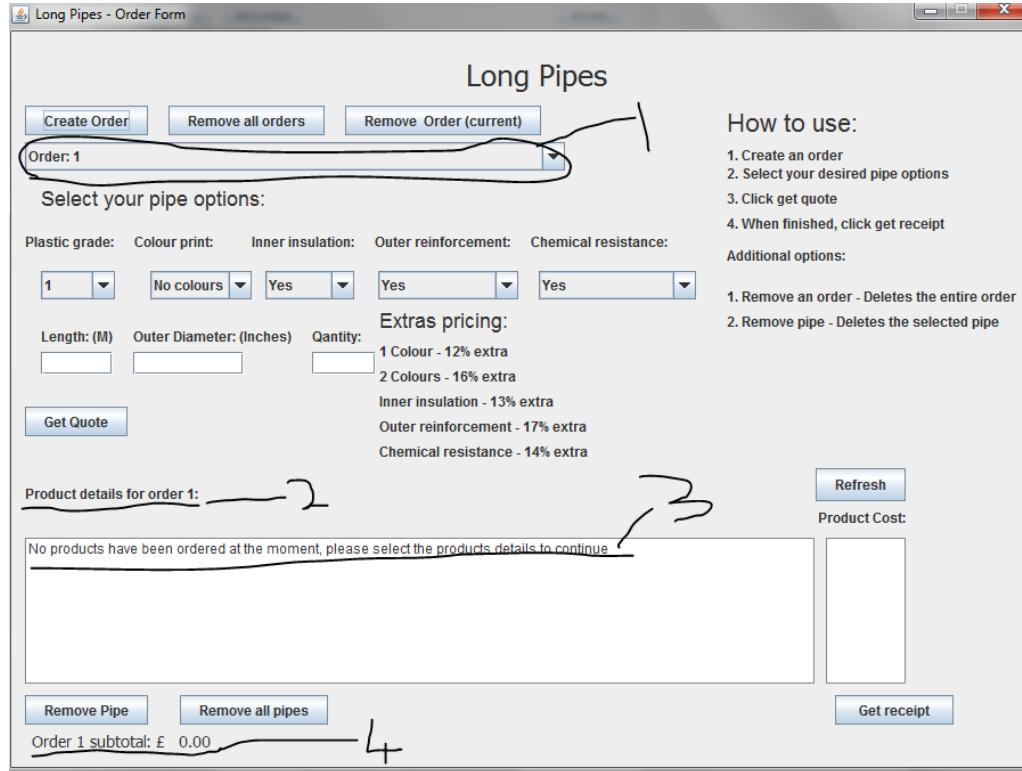
1. “Unable to show products details” should now show “Product details for order 1”
2. “Unable to calculate order subtotal” should now be changed to “Order 1 subtotal: £”
3. In the list of order, “No products have been ordered at the moment” should now be displayed.

On click response of create order button:



Here we can see that the correct message has been displayed. We now need to check to see what happens after the user clicks OK.

After on click of create order button

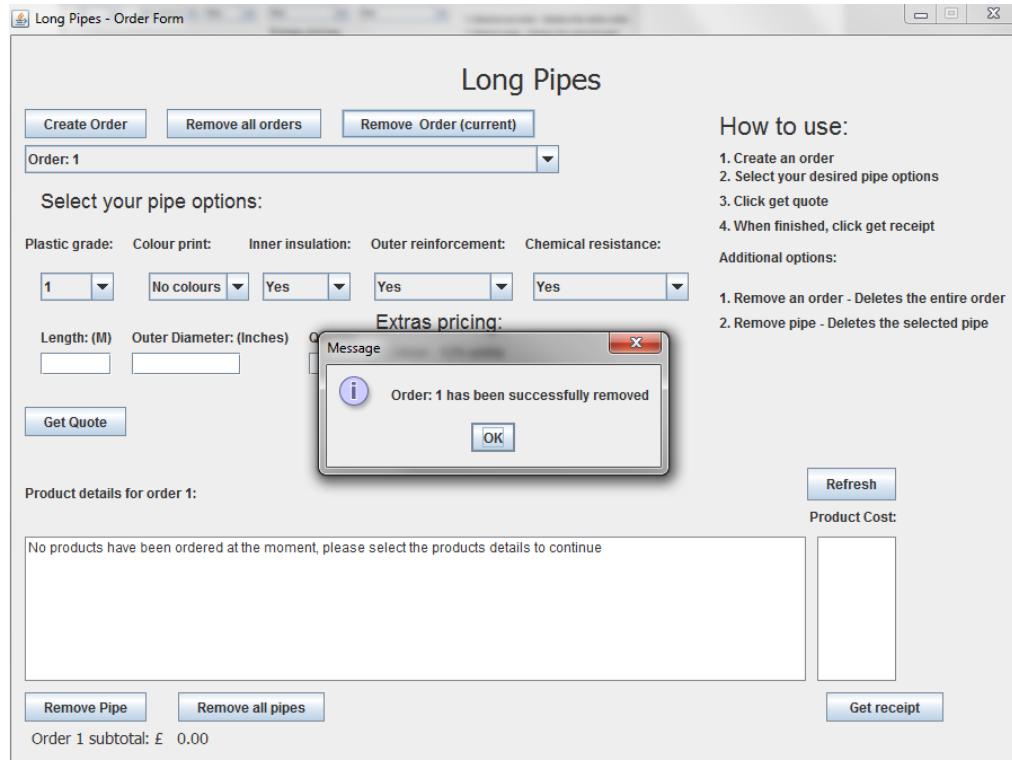


1. We can see that a new order has in fact been added to the list of orders.
2. We can see that product details label has been changed appropriately.
3. We can see that the list of products is showing that no products have been ordered at the moment
4. We can see that “Order 1 subtotal: £” is being displayed appropriately.

We can now say that all of our expectations have now been met so this test has PASSED.

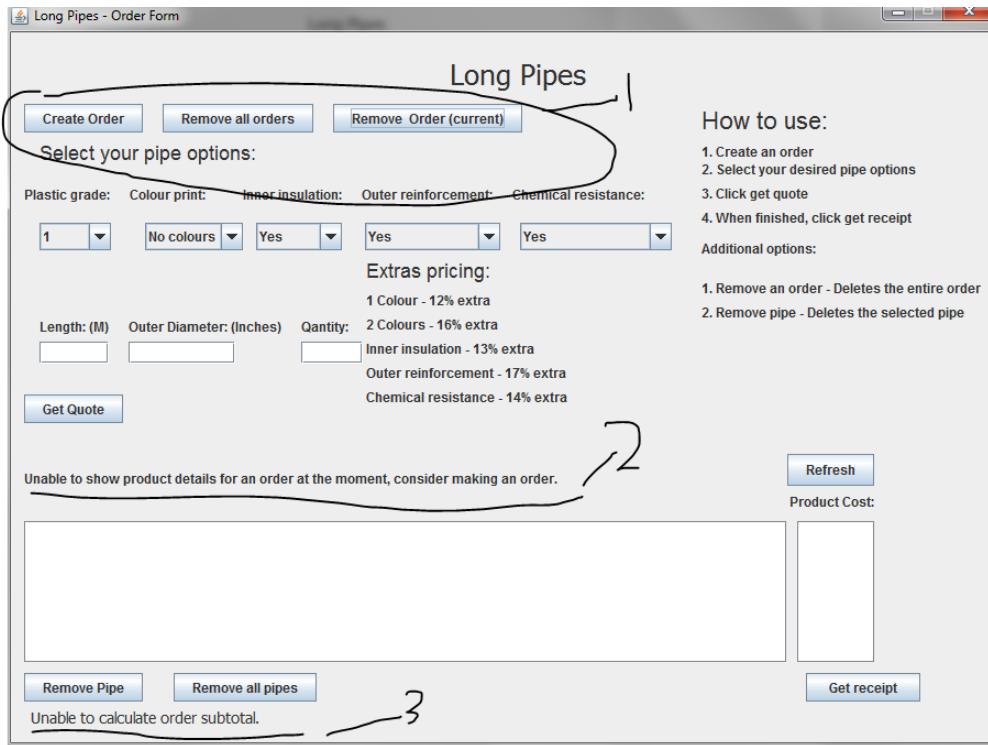
2. “Remove order” button deletes this order.

We expect, when the user has created an order, that the button should now successfully delete this order entirely and reset, if this was the last order to be made/ only one, to the original state of the program at run-time.



An appropriate error message has been showing, detailing that “order 1” the selected order has been removed”

After on click of “remove order (current)” button:



1. The drop-down menu, in this case, has been removed as Order 1 was the last order in the list.
2. Product 1 details has now been reset to the original state of the program –Initial run-time
3. “Order 1 subtotal” has been reset to its initial run-time setting

As these subtests have now passed, this test has passed.

### Test 3 – “Remove all orders” functionality

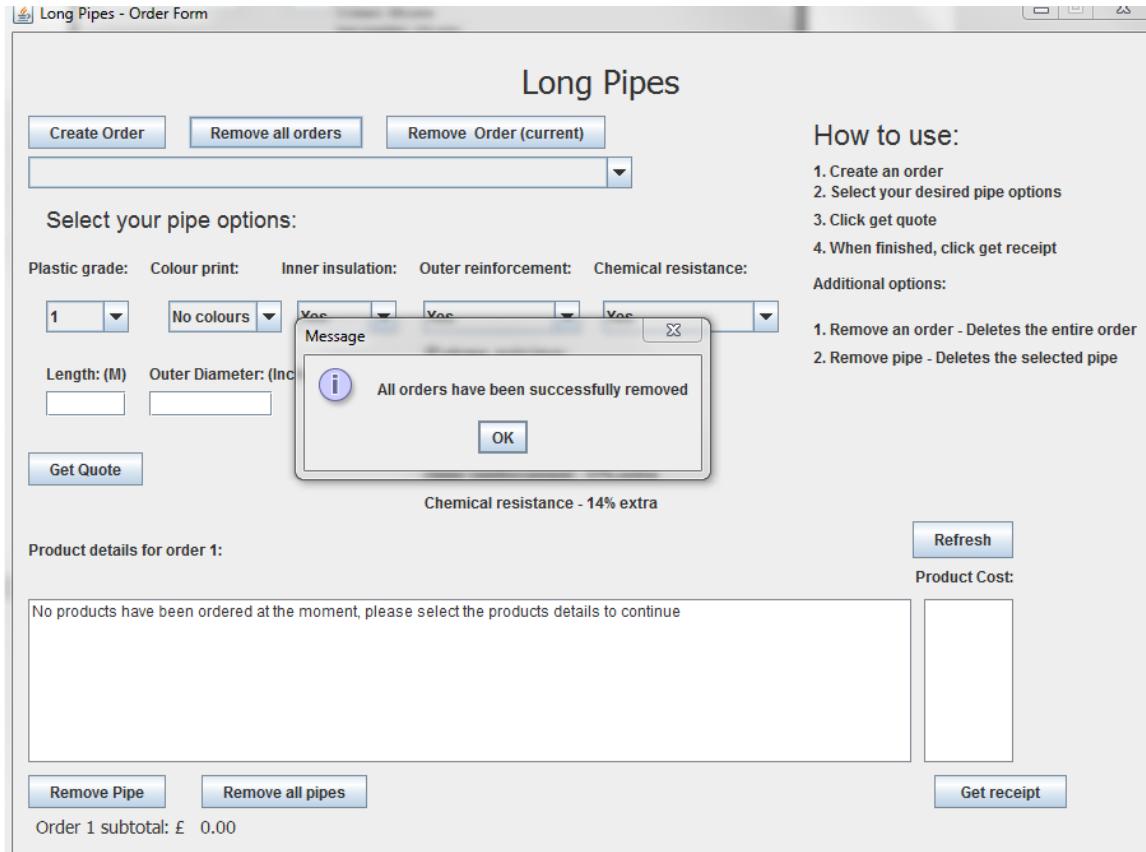
We expect that when the user has made more than 1 order, that all of these orders have been deleted and the program resets to its original initial run-time state as shown in the previous test when the last order has been removed.

Multiple orders have been made:

The screenshot shows the 'Long Pipes - Order Form' application window. At the top, there are three buttons: 'Create Order', 'Remove all orders', and 'Remove Order (current)'. Below these buttons is a dropdown menu labeled 'Order: 1' which lists 'Order: 1', 'Order: 2', and 'Order: 3'. To the right of the dropdown, there is a section titled 'How to use:' with four numbered steps: 1. Create an order, 2. Select your desired pipe options, 3. Click get quote, 4. When finished, click get receipt. Below this, there is a section titled 'Additional options:' with two items: 1. Remove an order - Deletes the entire order and 2. Remove pipe - Deletes the selected pipe. In the center of the form, there are input fields for 'Plastic grade:', 'Colour print:', 'Inner insulation:', 'Outer reinforcement:', and 'Chemical resistance:', each with dropdown menus. To the right of these fields is a section titled 'Extras pricing:' with options: 1 Colour - 12% extra, 2 Colours - 16% extra, Inner insulation - 13% extra, Outer reinforcement - 17% extra, and Chemical resistance - 14% extra. Below these sections are buttons for 'Get Quote' and 'Refresh'. On the left, there is a section titled 'Product details for order 1:' containing a message: 'No products have been ordered at the moment, please select the products details to continue'. On the right, there is a section titled 'Product Cost:' with a large empty box. At the bottom of the form, there are buttons for 'Remove Pipe' and 'Remove all pipes' on the left, and 'Get receipt' on the right. A message 'Order 1 subtotal: £ 0.00' is displayed at the bottom center.

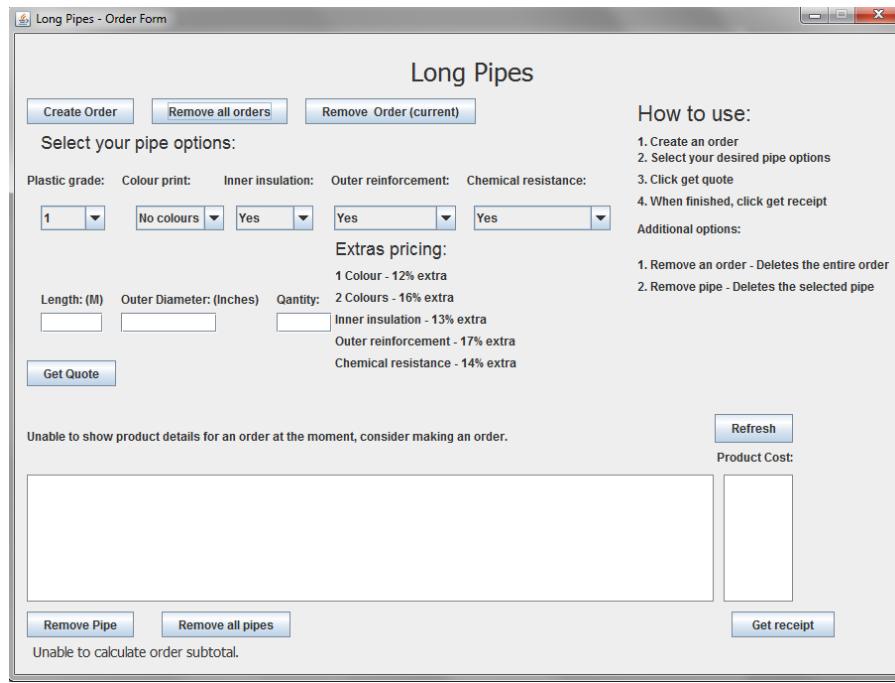
Here we can see that multiple orders have been made, we shall now attempt to remove all orders.

Test 4 - Response of remove all orders functionality



The appropriate message has been displayed to the user, we shall now see what happens after the user clicks ok.

After OK on click of “All orders removed” message:



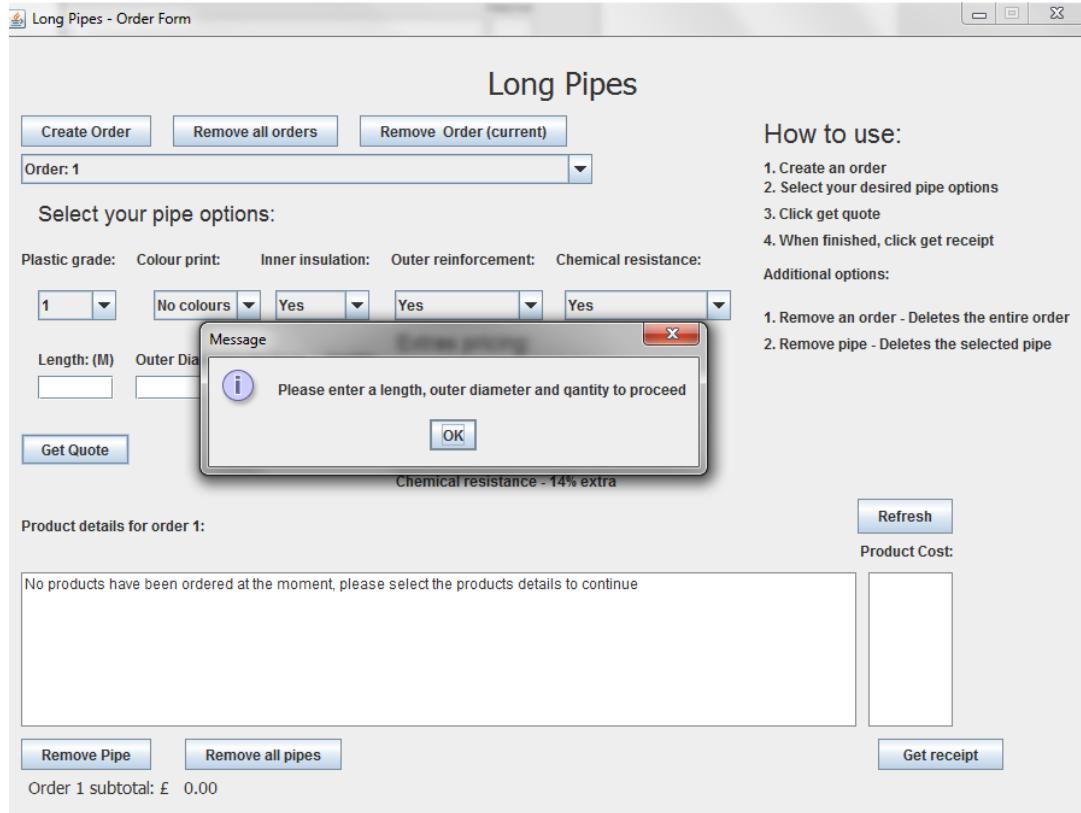
The program has now reset to its original state.

This test has PASSED.

#### Test 4 – Get Quote functionality

At this point in the program, the as an order has been made the program is now ready to start accepting orders of different products. We shall now check the Get Quote button functionality.

Response from Get Quote button:



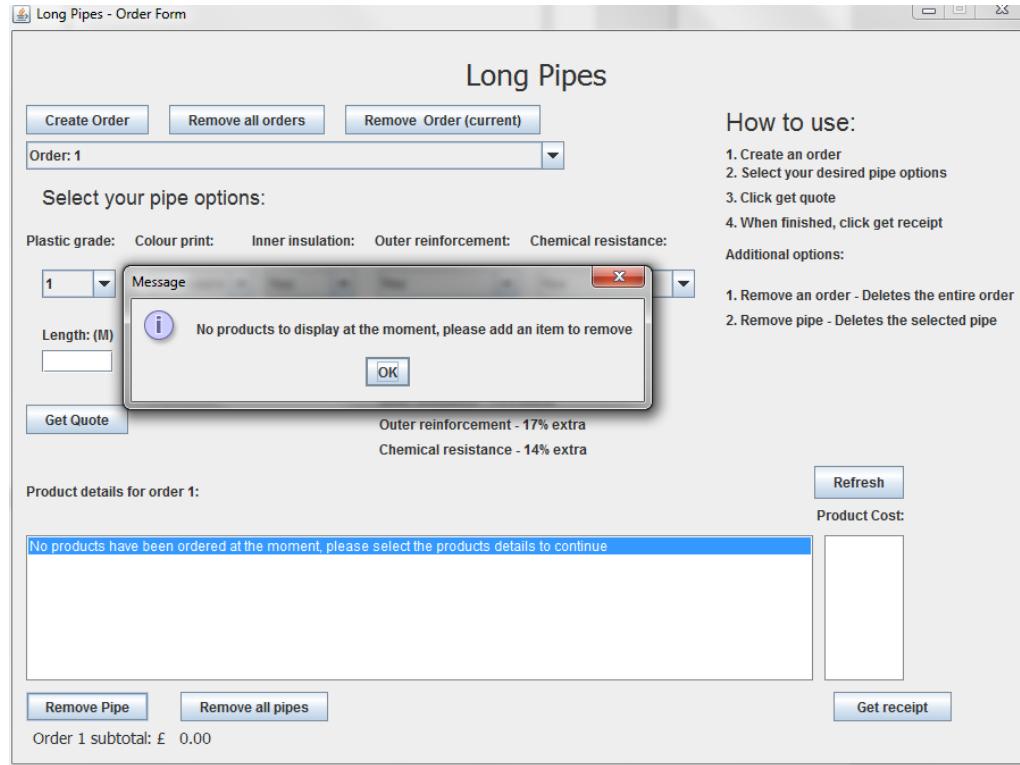
As no information has been placed in the length, outer diameter and quantity fields. This is the appropriate error message received by the user and demonstrates that the program is ready for new products to be found and added to the order.

This is what we expect at this stage so this test has PASSED.

Test 5 – Remove pipe button functionality

At first an order has only been made, there are no products in the order at the moment and therefore the remove pipe button should display the appropriate error message.

### Response from Remove pipe button

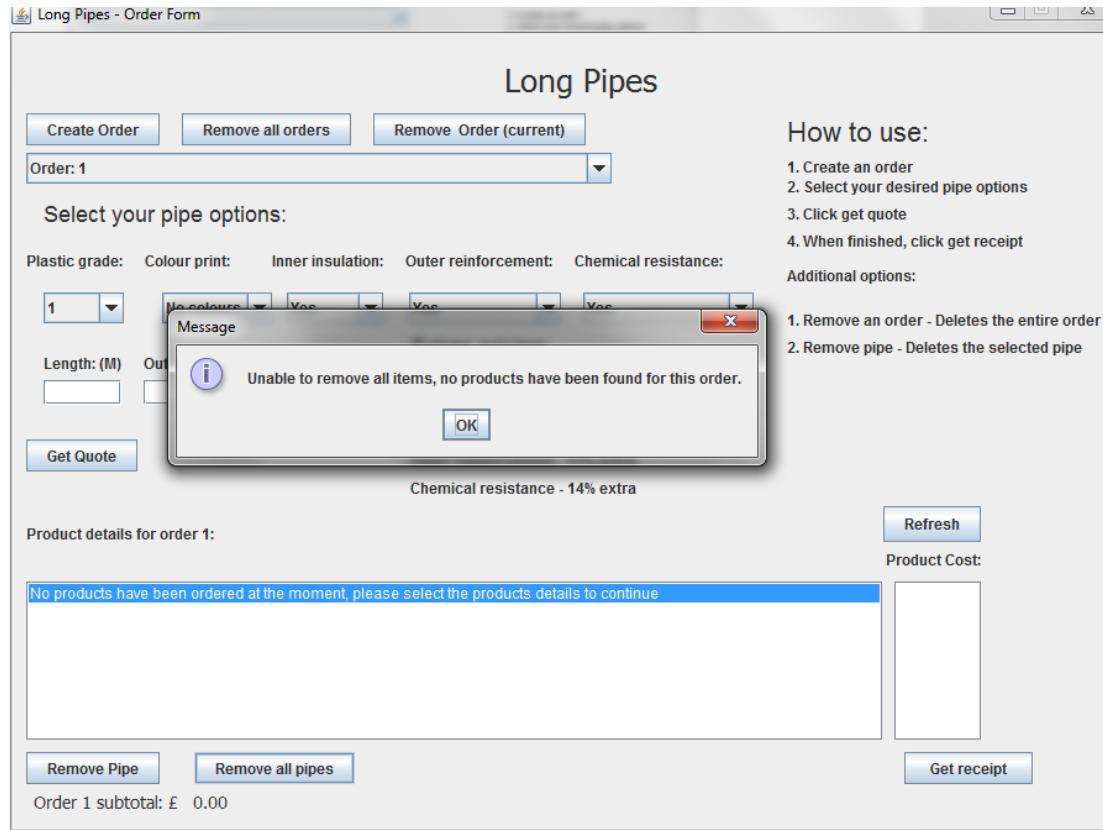


This is what we expect to see so this test has PASSED.

### Test 6 – Remove all pipes

As explained above, no products are in the order at the moment so that remove all pipes button should respond with the appropriate error message.

Response from Remove all pipes button:

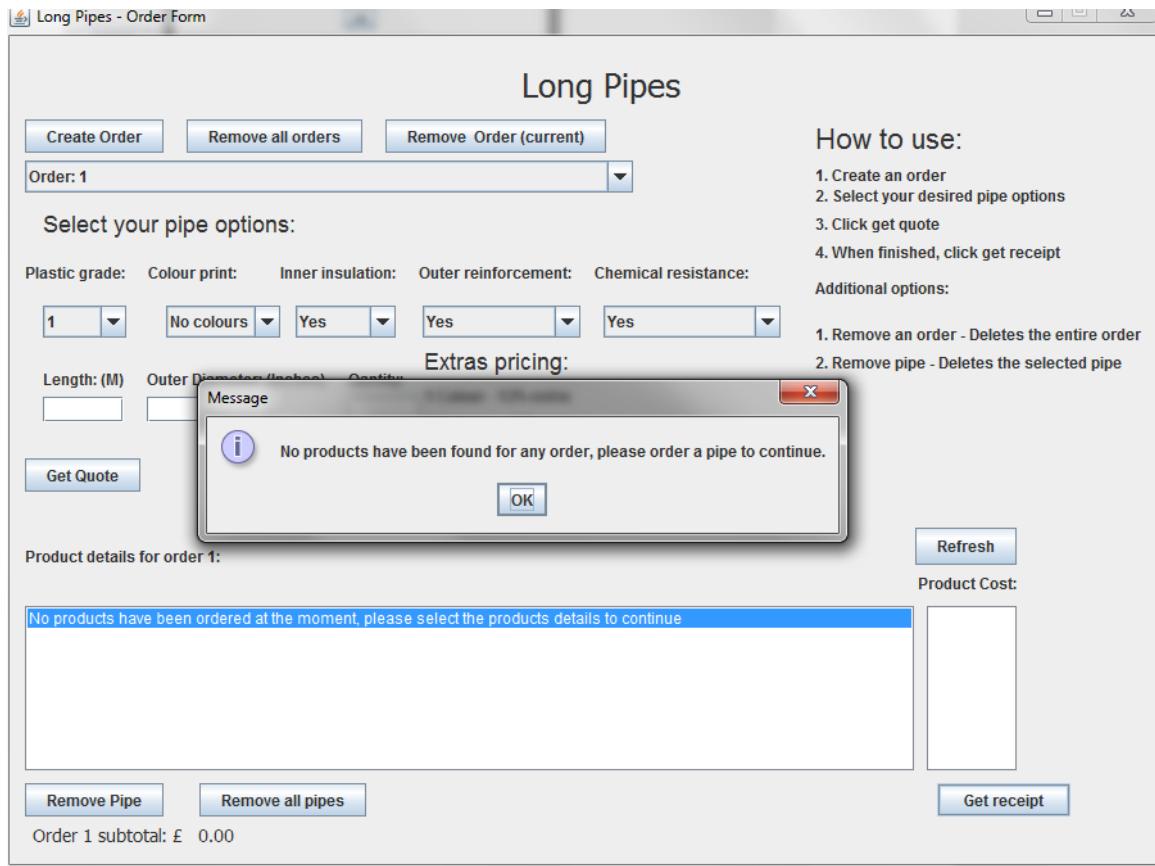


This is what we expected so this test has PASSED.

Test 7 – Get Receipt button functionality

As no products have been made, the program is unable to produce a receipt at this stage. Get Receipt button should respond with the appropriate error message to reflect this.

Response from Get Receipt button:



This is what we expected so this test has PASSED.

#### Area 2 – Create Order button and various functionality from other buttons

1. Create order button makes a new order and this is shown - PASSED
2. Remove order button deletes this order - PASSED
3. Remove all order button removes multiple orders - PASSED
4. Get Quote functionality – Appropriate error message (On click) - PASSED
5. Remove Pipe functionality – Appropriate error message (On click) - PASSED
6. Remove All Pipes functionality – Appropriate error message (On click) - PASSED
7. Get receipt functionality – Appropriate error message (On click) - PASSED

7/7 passes so this section has PASSED.

### Area 3 – Adding a Pipe

In this area we are looking for the following results:

1. Finding and adding the correct type of pipe that is supplied
2. Appropriate error checking at all levels are handled appropriately when adding a pipe
3. The pipe is added
4. The Pipe cost is shown
5. The receipt screen now shows

### Test 1 – Finding and adding a pipe that is supplied

In this test scenario, we will be using valid test data in order to show that the program does only respond to adding a pipe if it is supplied by Long Pipe ltd, otherwise an appropriate error message is shown.

These are the following types of pipe that Long Pipes supply.

TYPE	PLASTIC'S GRADE	COLOUR PRINT			INNER INSULATION	OUTER REINFORCEMENT	CHEMICAL RESISTANCE
		0	1	2			
I	1 - 3	YES	NO	NO	NO	NO	YES/NO
II	2 - 4	NO	YES	NO	NO	NO	YES/NO
III	2 - 5	NO	NO	YES	NO	NO	YES/NO
IV	2 - 5	NO	NO	YES	YES	NO	YES/NO
V	3 - 5	NO	NO	YES	YES	YES	YES/NO

In our first subtest, we will check if Pipe Grade 1, with no Color print, no inner insulation, no outer reinforcement and no chemical resistance is found and if so, display an appropriate message for this and add the pipe to their user's selected order.

Entered information before Get Quote has been clicked:

**Long Pipes - Order Form**

## Long Pipes

**Select your pipe options:**

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1	No colours	No	No	No
---	------------	----	----	----

Length: (M) Outer Diameter: (Inches) Quantity:

1	1	1
---	---	---

**Extras pricing:**

- 1 Colour - 12% extra
- 2 Colours - 16% extra
- Inner insulation - 13% extra
- Outer reinforcement - 17% extra
- Chemical resistance - 14% extra

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

**Additional options:**

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

**Product details for order 1:**

No products have been ordered at the moment, please select the products details to continue

**Product Cost:**

**Buttons:**

- Get Quote
- Refresh
- Remove Pipe
- Remove all pipes
- Get receipt

Order 1 subtotal: £ 0.00

Response from Get Quote button:

**Long Pipes - Order Form**

## Long Pipes

**Select your pipe options:**

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1	No colours	No	No	No
---	------------	----	----	----

Length: (M) Outer Diameter:

1	1
---	---

**Message**

Pipe has been sucessfully added, please continue.

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

**Additional options:**

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

**Product details for order 1:**

No products have been ordered at the moment, please select the products details to continue

**Product Cost:**

**Buttons:**

- Get Quote
- Refresh
- Remove Pipe
- Remove all pipes
- Get receipt

Order 1 subtotal: £ 0.00

Here we can see that the appropriate message has been displayed i.e. a type of pipe has been found and added, after the user clicks ok, the type of pipe that they requested should now show in the product list for Order 1.

Response of Get Quote after user clicks OK:

The screenshot shows a Windows application window titled "Long Pipes - Order Form". The main title bar says "Long Pipes". The window contains several input fields and buttons. At the top left are buttons for "Create Order", "Remove all orders", and "Remove Order (current)". Below that is a dropdown menu labeled "Order: 1". To the right of the dropdown is a section titled "How to use:" with a numbered list of four steps: 1. Create an order, 2. Select your desired pipe options, 3. Click get quote, 4. When finished, click get receipt. Underneath this is another section titled "Additional options:" with two items: 1. Remove an order - Deletes the entire order and 2. Remove pipe - Deletes the selected pipe. In the center-left area, there's a section titled "Select your pipe options:" with dropdown menus for "Plastic grade" (set to 1), "Colour print" (No colours), "Inner insulation" (Yes), "Outer reinforcement" (Yes), and "Chemical resistance" (Yes). Below these are input fields for "Length: (M)", "Outer Diameter: (Inches)", and "Quantity". To the right of these fields is a "Extras pricing:" section with options like "1 Colour - 12% extra", "2 Colours - 16% extra", "Inner insulation - 13% extra", "Outer reinforcement - 17% extra", and "Chemical resistance - 14% extra". A "Get Quote" button is located below the extras section. On the right side of the window, there's a "Refresh" button and a "Product Cost:" field containing "2.35". At the bottom left are buttons for "Remove Pipe" and "Remove all pipes". At the very bottom left is the text "Order 1 cost: 2.35".

We can now see that type 1 was found as predicted earlier as the user entered in pipe properties that corresponded with Pipe type 1 so therefore this pipe was found, as it is supplied by Long Pipes, and the Pipe has now been added to the product list and shown above.

This test has passed.

The product details for the pipe is show as well in the product list as well as cost.

Next we need show that you the program does not supply a pipe of grade 1, no colors but with inner insulation and no outer reinforcement and no chemical resistance.

**Long Pipes - Order form**

## Long Pipes

**Create Order    Remove all orders    Remove Order (current)**

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

2	No colours	Yes	No	No
---	------------	-----	----	----

Length: (M)   Outer Diameter: (Inches)   Quantity: 1

**Extras pricing:**

- 1 Colour - 12% extra
- 2 Colours - 16% extra
- Inner insulation - 13% extra
- Outer reinforcement - 17% extra
- Chemical resistance - 14% extra

**Get Quote    Refresh    Product Cost:**

No products have been ordered at the moment, please select the products details to continue

**Remove Pipe    Remove all pipes    Get receipt**

Order 1 subtotal: £ 0.00

Response of Get Quote button:

**Long Pipes**

**Create Order    Remove all orders    Remove Order (current)**

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

2	No colours	Yes	No	No
---	------------	-----	----	----

Length: (M)   Outer Diameter: (Inches)   Quantity: 1

**Extras pricing:**

- 1 Colour - 12% extra
- 2 Colours - 16% extra

**Get Quote**

**Message**

We do not stock this type of pipe, please try again.

**OK**

**Product details for order 1:**

No products have been ordered at the moment, please select the products details to continue

**Refresh    Product Cost:**

**Remove Pipe    Remove all pipes    Get receipt**

Order 1 subtotal: £ 0.00

We can see that this type of pipe is not supplied by Long Pipes therefore the appropriate error message has been displayed. The type of pipe that this program would try and match this properties with is type 1 but failed.

We shall now try and add a pipe of grade 5, 2 colors with inner insulation and outer reinforcement but with no chemical resistance.

This pipe type should be found and added.

The screenshot shows the 'Long Pipes - Order Form' application window. At the top, there are buttons for 'Create Order', 'Remove all orders', and 'Remove Order (current)'. A dropdown menu labeled 'Order: 1' is open. Below these, a section titled 'Select your pipe options:' contains fields for Plastic grade (5), Colour print (2 colours), Inner insulation (Yes), Outer reinforcement (Yes), and Chemical resistance (No). To the right, a 'How to use:' section lists steps 1-4 for creating an order. Under 'Additional options:', it lists 'Remove an order - Deletes the entire order' and 'Remove pipe - Deletes the selected pipe'. In the center, there's an 'Extras pricing:' section with options for '1 Colour - 12% extra', '2 Colours - 16% extra', and 'Inner insulation - 13% extra'. A 'Get Quote' button is located to the left of this section. A 'Message' dialog box is overlaid on the form, stating 'Pipe has been successfully added, please continue.' with an 'OK' button. On the right side, there are 'Refresh', 'Product Cost', and 'Get receipt' buttons. At the bottom, there are 'Remove Pipe' and 'Remove all pipes' buttons, and a note 'Order 1 subtotal: £ 0.00'.

We can see that a type of pipe has been found and therefore has been added. We shall see the results below.

**Long Pipes - Order Form**

## Long Pipes

**Create Order      Remove all orders      Remove Order (current)**

**Order: 1**

**Select your pipe options:**

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1    No colours    Yes    Yes    Yes

Length: (M)    Outer Diameter: (Inches)    Quantity:

Get Quote

**Extras pricing:**

- 1 Colour - 12% extra
- 2 Colours - 16% extra
- Inner insulation - 13% extra
- Outer reinforcement - 17% extra
- Chemical resistance - 14% extra

**Product details for order 1:**

Pipe grade: 5, Length: 1.00, Outer diameter: 1.0, Qantity: 1, Chemical resistance: No, Colours: 2, Insulated: Yes, Reinforced: Yes

**Refresh**

**Product Cost:**

8.15

**Remove Pipe      Remove all pipes      Get receipt**

Order 1 cost: 8.15

The program has found a pipe of type 5 and has added this pipe to the product list as expected.

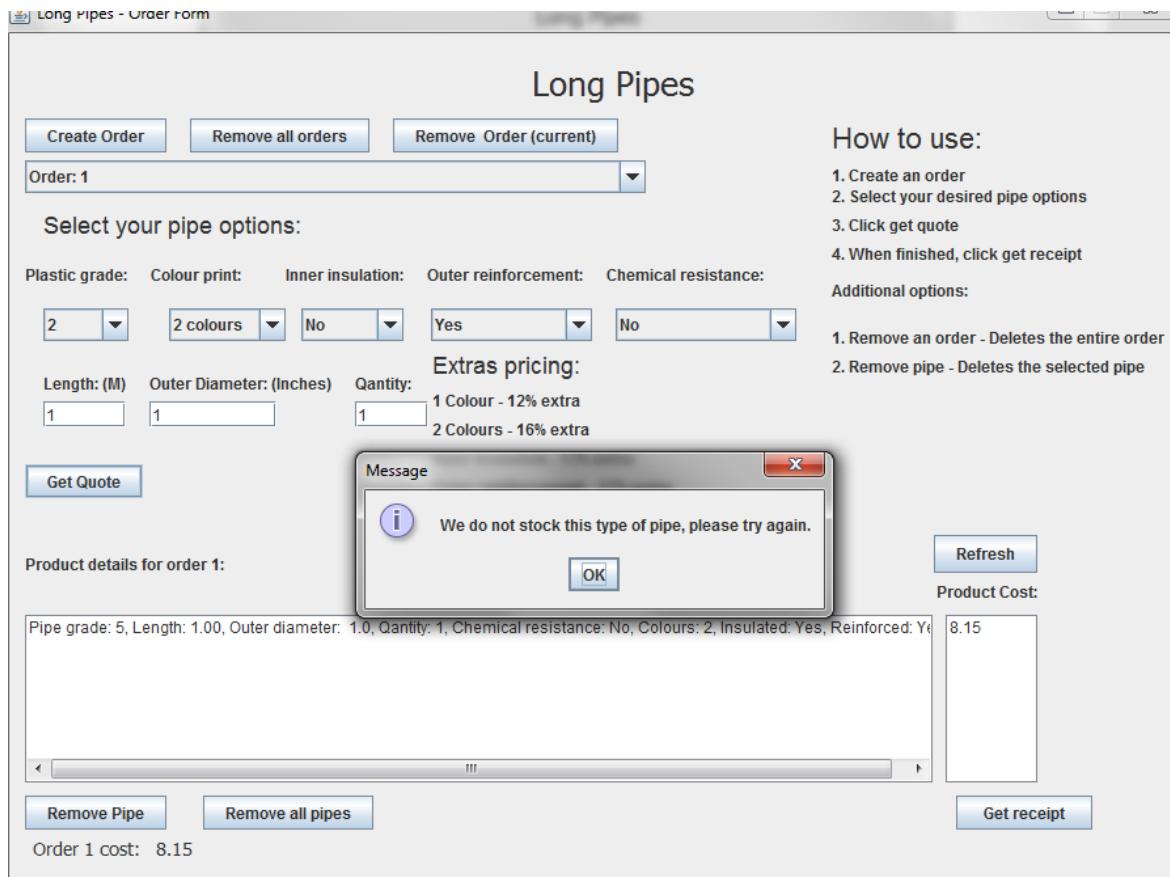
This test has passed.

Next we shall test to see that a requested pipe with options, plastic grade 2, 1 color and inner insulation but no outer reinforcement and no chemical resistance should not be added as this is not supplied by long pipes.

This is the appropriate error message as Long Pipes do not supply this type of pipe. The type of pipe that this program would try and match this properties with is type 2 but failed.

This test has passed.

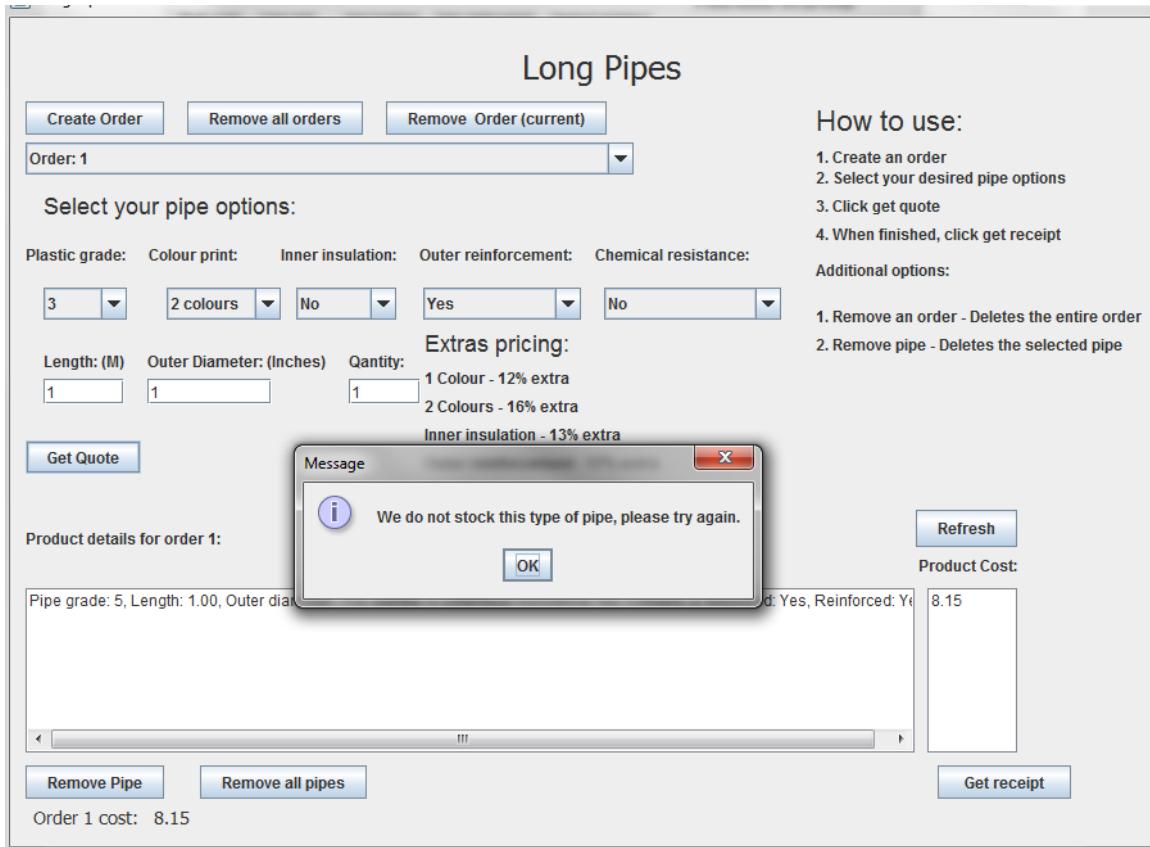
We will now check that if the user enters pipe grade 2 with 2 colors, no inner insulation and outer reinforcement with no chemical resistance should be rejected as this is not supplied by Long Pipes.



This is the appropriate error message as Long Pipes do not supply this type of pipe. The type of pipe that this program would try and match this properties with is type 3 but failed.

This test has passed.

Next we will check if getting a quote for a pipe with properties grade 3, 2 colors, no inner insulation but has outer reinforcement and no chemical resistance is no supplied by Long Pipes.



This is the appropriate error message as Long Pipes do not supply this type of pipe. The type of pipe that this program would try and match this properties with is type 4 but failed.

This test has passed.

Lastly, we shall see that a pipe with properties grade 2, with 2 colors, inner insulation, outer reinforcement and no chemical resistance should be rejected as Long Pipes only supplies pipes of type 5 with a plastic grade of 3 or above.

## Long Pipes

[Create Order](#)   [Remove all orders](#)   [Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade:  Colour print:  Inner insulation:  Outer reinforcement:  Chemical resistance:

Length: (M)  Outer Diameter: (Inches)  Quantity:

**Extras pricing:**

- 1 Colour - 12% extra
- 2 Colours - 16% extra

[Get Quote](#)

Product details for order 1:

Pipe grade: 5, Length: 1.00, Outer diameter: 1.0, Qantity: 1, Chemical resistance: No, Colours: 2, Insulated: Yes, Reinforced: Yes

[Remove Pipe](#)   [Remove all pipes](#)

Order 1 cost: 8.15

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

[Refresh](#)

Product Cost:

[Get receipt](#)

We do not stock this type of pipe, please try again.

This is the appropriate error message as Long Pipes do not supply this type of pipe. The type of pipe that this program would try and match this properties with is type 5 but failed.

This test has passed.

Test 3 – Finding and adding the correct supplied pipe summary

7/7 passes – This test has now passed.

Test 3 - Appropriate error checking at all levels are handled appropriately when adding a pipe

Now we will check that when the user enters no information into any fields, an appropriate error message is displayed.

The screenshot shows the 'Long Pipes' application interface. At the top, there are three buttons: 'Create Order', 'Remove all orders', and 'Remove Order (current)'. Below them is a dropdown menu labeled 'Order: 1'. To the right, there is a 'How to use:' section with numbered steps and additional options. In the center, there's a form for selecting pipe options, including dropdowns for Plastic grade (set to 1), Colour print (No colours), Inner insulation (No), Outer reinforcement (No), and Chemical resistance (No). Below these are input fields for Length: (M), Outer Diameter: (Inches), and Quantity. To the right of these fields is an 'Extras pricing:' section with options like '1 Colour - 12% extra', '2 Colours - 16% extra', 'Inner insulation - 13% extra', and 'Outer reinforcement - 17% extra'. A 'Get Quote' button is located below the pipe options. A message box titled 'Message' is overlaid on the interface, displaying the error message 'Please enter a length, outer diameter and quantity to proceed' with an 'OK' button. At the bottom left, there are buttons for 'Remove Pipe' and 'Remove all pipes', and at the bottom right, there are buttons for 'Refresh', 'Product Cost:', 'Get receipt', and 'Order 1 cost: 0.00'.

We can see that there is no information in any of the fields for length, quantity and outer diameter therefore the error message has been displayed.

This test has passed.

Next we will check our type errors that can occur. In this application the type errors that may occur include:

- . Entering a string/character (word or letter) in any of the fields. The length and outer diameter fields require a rational number or integer whereas the quantity field only allows an integer.

We shall now check that the appropriate errors are handled appropriately.

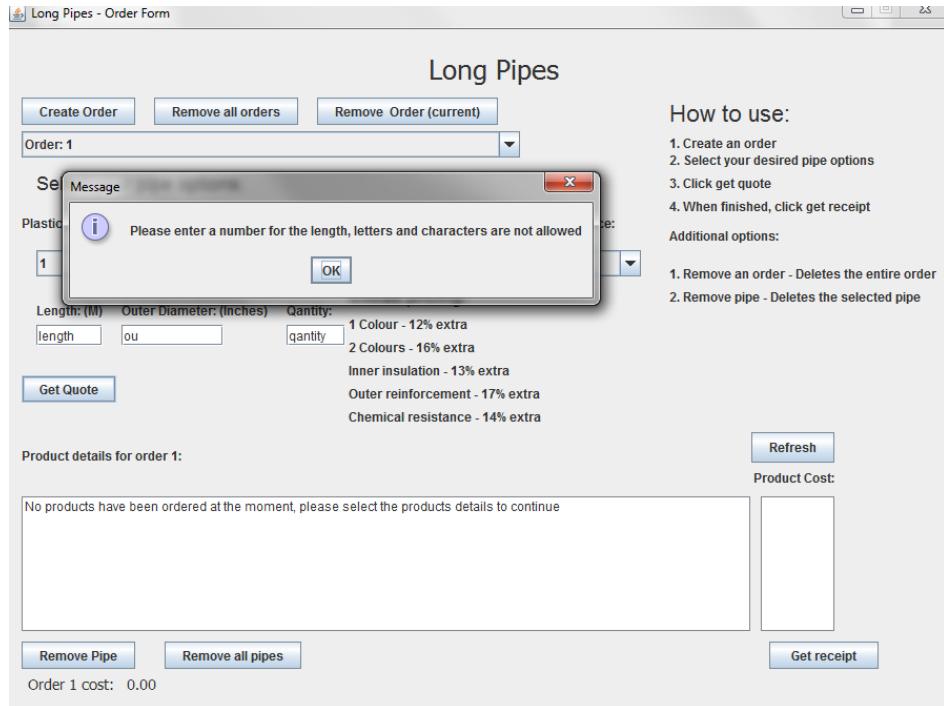
If the user enters a letter/word for each field, the program should respond appropriately.

The screenshot shows a web-based application titled "Long Pipes". The interface includes:

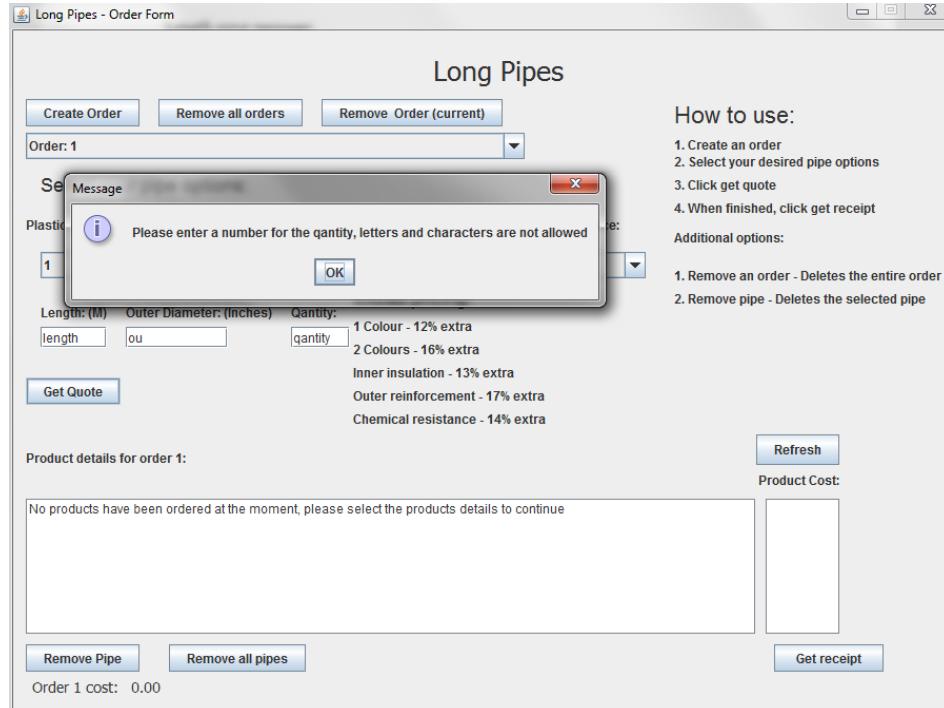
- Buttons:** Create Order, Remove all orders, Remove Order (current), Get Quote, Refresh, Get receipt.
- Text Input:** Order: 1
- Section:** Select your pipe options:
  - Plastic grade: 1
  - Colour print: No colours
  - Inner insulation: No
  - Outer reinforcement: No
  - Chemical resistance: No
- Text Input:** Length: (M) length, Outer Diameter: (Inches) ou, Quantity: quantity
- Section:** Extras pricing:
  - 1 Colour - 12% extra
  - 2 Colours - 16% extra
  - Inner insulation - 13% extra
  - Outer reinforcement - 17% extra
  - Chemical resistance - 14% extra
- Section:** Product details for order 1:
  - No products have been ordered at the moment, please select the products details to continue
- Buttons:** Remove Pipe, Remove all pipes, Get receipt.
- Text:** Order 1 cost: 0.00
- Section:** How to use:
  - Create an order
  - Select your desired pipe options
  - Click get quote
  - When finished, click get receipt
- Section:** Additional options:
  - Remove an order - Deletes the entire order
  - Remove pipe - Deletes the selected pipe
- Text:** Product Cost: (empty box)

After clicking get quote button:

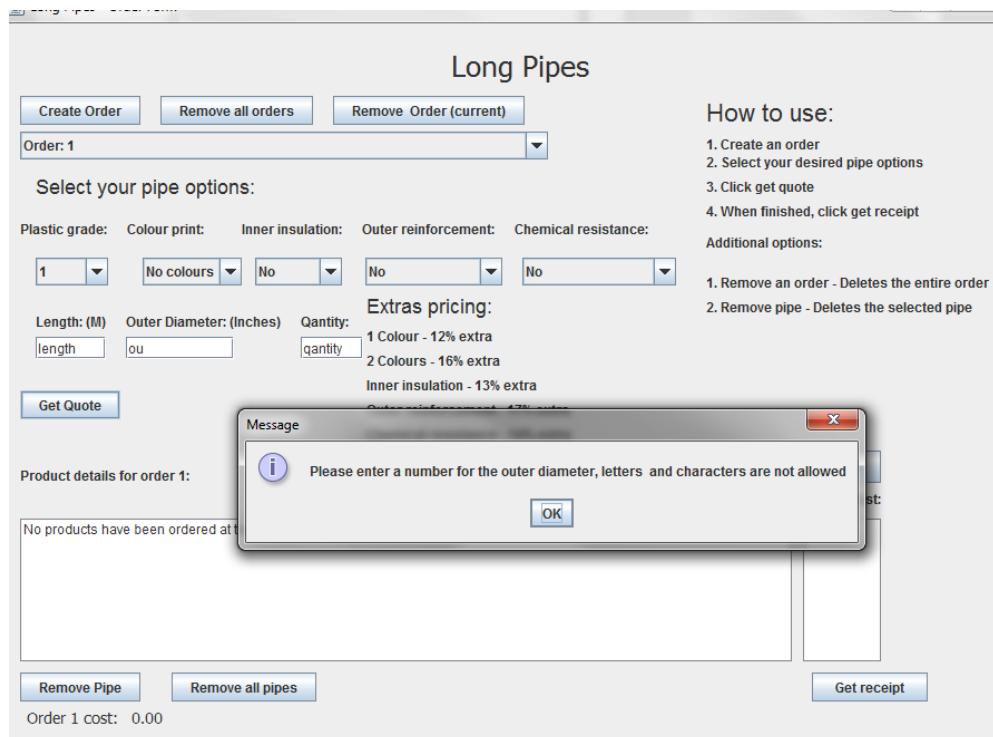
Length error message:



Quantity error message:



Outer diameter error message:



All of these firsts stages of the subtests behave as expected.

This test has passed.

Next we shall check that the program responds appropriately when the user enters a word in only 1 field and appropriately valid types in the other fields.

### Long Pipes

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade:	Colour print:	Inner insulation:	Outer reinforcement:	Chemical resistance:
<input type="button" value="1"/>	<input type="button" value="No colours"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>

Length: (M) Outer Diameter: (Inches) Quantity:

<input type="text" value="1"/>	<input type="text" value="ou"/>	<input type="text" value="1"/>
--------------------------------	---------------------------------	--------------------------------

**Extras pricing:**

1 Colour - 12% extra
2 Colours - 16% extra
Inner insulation - 13% extra
Outer reinforcement - 17% extra
Chemical resistance - 14% extra

[Get Quote](#) [Refresh](#) [Product Cost:](#)

Product details for order 1:

No products have been ordered at the moment, please select the products details to continue

[Remove Pipe](#) [Remove all pipes](#) [Get receipt](#)

Order 1 cost: 0.00

After click of Get Quote;

The screenshot shows the 'Long Pipes' order form with a modal message box in the foreground. The message box has an information icon and the text: "Please enter a number for the outer diameter, letters and characters are not allowed". It also contains an "OK" button.

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 1

Please enter a number for the outer diameter, letters and characters are not allowed

Length: (M) Outer Diameter: (Inches) Quantity:

<input type="text" value="1"/>	<input type="text" value="ou"/>	<input type="text" value="1"/>
--------------------------------	---------------------------------	--------------------------------

**Extras pricing:**

1 Colour - 12% extra
2 Colours - 16% extra
Inner insulation - 13% extra
Outer reinforcement - 17% extra
Chemical resistance - 14% extra

[Get Quote](#) [Refresh](#) [Product Cost:](#)

Product details for order 1:

No products have been ordered at the moment, please select the products details to continue

[Remove Pipe](#) [Remove all pipes](#) [Get receipt](#)

Order 1 cost: 0.00

We can see that this is the expected response.

This test has passed.

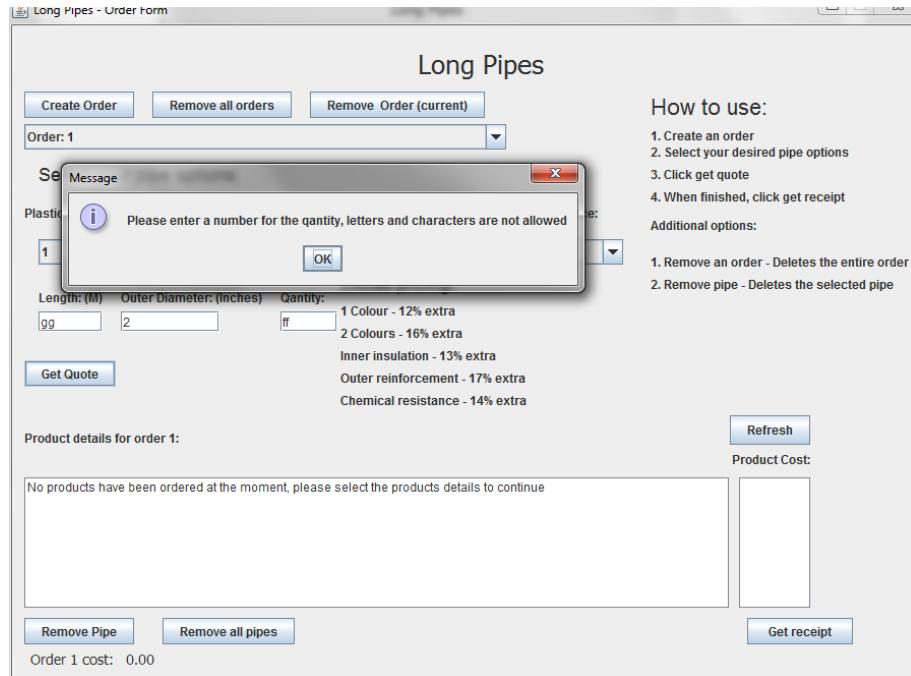
We shall now check for when the user enters wrong types for 2 random fields.

The screenshot shows the 'Long Pipes' application interface. At the top, there are three buttons: 'Create Order', 'Remove all orders', and 'Remove Order (current)'. Below them is a dropdown menu set to 'Order: 1'. A section titled 'Select your pipe options:' contains several dropdown menus and checkboxes. The 'Plastic grade:' dropdown is set to '1'. The 'Colour print:' dropdown has 'No colours' selected. The 'Inner insulation:' and 'Outer reinforcement:' dropdowns both have 'No' selected. The 'Chemical resistance:' dropdown has 'No' selected. Below these are input fields for 'Length: (M)' (containing 'gg'), 'Outer Diameter: (Inches)' (containing '2'), and 'Quantity:' (containing 'ff'). To the right of these fields is a 'Extras pricing:' table with rows for '1 Colour - 12% extra', '2 Colours - 16% extra', 'Inner insulation - 13% extra', 'Outer reinforcement - 17% extra', and 'Chemical resistance - 14% extra'. Below the input fields are two buttons: 'Get Quote' and 'Refresh'. To the right of the input fields is a 'How to use:' section with numbered steps and additional options. At the bottom left are 'Remove Pipe' and 'Remove all pipes' buttons, and at the bottom center is a button for 'Get receipt'. The message 'Product details for order 1:' is displayed above a large text area containing the message 'No products have been ordered at the moment, please select the products details to continue'. At the very bottom, it says 'Order 1 cost: 0.00'.

Response of Get Quote after click:

The screenshot shows the same 'Long Pipes' application interface as the previous one, but with a modal dialog box in the foreground. The dialog box has a title 'Message' and a message 'Please enter a number for the length, letters and characters are not allowed'. It has an 'OK' button. The rest of the interface is visible in the background, including the 'Create Order' button, 'Remove all orders' button, 'Remove Order (current)' button, and the 'Order: 1' dropdown. The 'Select your pipe options:' section and 'Extras pricing:' table are also visible. The 'How to use:' section and other buttons at the bottom are present. The message 'Product details for order 1:' is displayed above a large text area containing the message 'No products have been ordered at the moment, please select the products details to continue'. At the very bottom, it says 'Order 1 cost: 0.00'.

This is what we expect from the length field as gg and been entered as is not permitted.

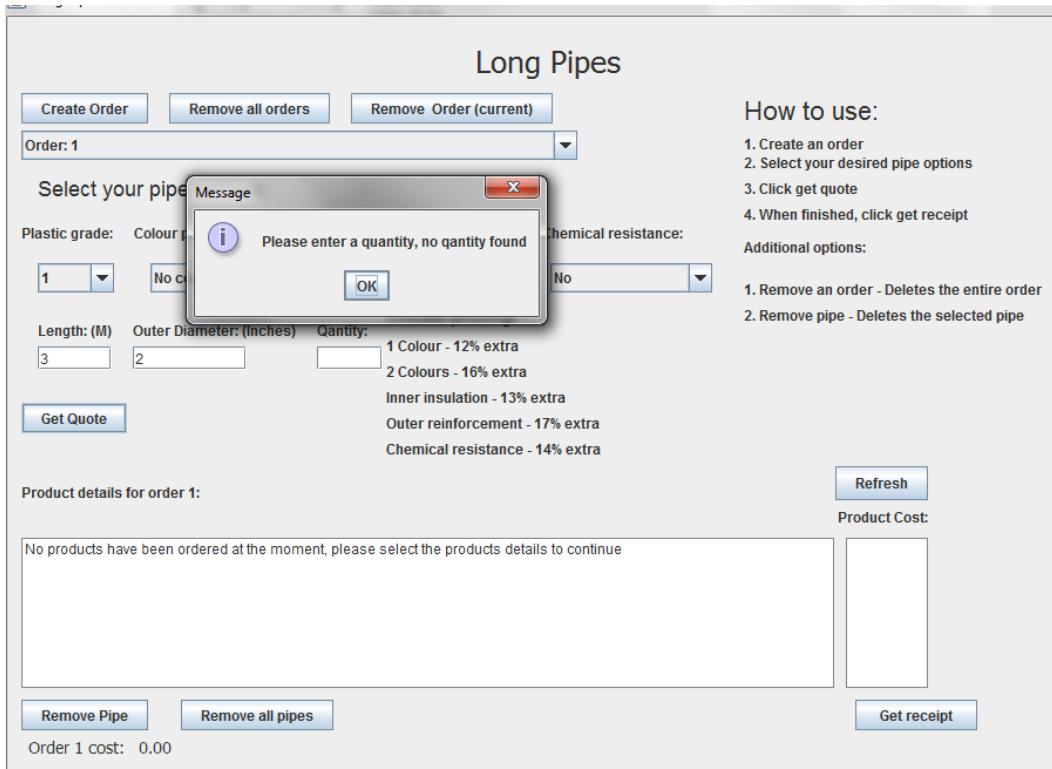


This is what we expect from the quantity field as if and been entered as is not permitted.

Both of these subtests pass and demonstrate how the program responds to random combinational type errors so we can now continue.

This test has passed.

Next we will check to see how the program responds when the user does not enter any information for a field but the 2 other fields are full.



This is what we expect to see as no data has been entered into to the quantity field.

This test has passed.

Next we will check for the length and outer diameter fields that if the user accidentally types in a rational number 3.4, 2.9 etc. and include a letter or word that an appropriate error message is displayed.

**Long Pipes**

**Create Order    Remove all orders    Remove Order (current)**

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

Length: (M) Outer Diameter: (Inches) Quantity:

1 No colours No No No  
3.hhh 2 1

**Extras pricing:**

1 Colour - 12% extra  
2 Colours - 16% extra  
Inner insulation - 12% extra

**Get Quote**

**Message**

Please enter a number for the length, letters and characters are not allowed

OK

Product details for order

No products have been ordered at the moment, please select one products details to continue

**Refresh**

**Product Cost:**

**Get receipt**

**Remove Pipe    Remove all pipes**

Order 1 cost: 0.00

The screenshot shows a software application window titled 'Long Pipes'. At the top, there are three buttons: 'Create Order', 'Remove all orders', and 'Remove Order (current)'. Below this is a dropdown menu set to 'Order: 1'. A section titled 'Select your pipe options:' contains five dropdown menus for 'Plastic grade', 'Colour print', 'Inner insulation', 'Outer reinforcement', and 'Chemical resistance', all currently set to 'No'. Below this are fields for 'Length: (M)' (containing '3.hhh'), 'Outer Diameter: (Inches)' (containing '2'), and 'Quantity' (containing '1'). To the right of these fields is a 'Extras pricing:' section with two options: '1 Colour - 12% extra' and '2 Colours - 16% extra', with 'Inner insulation - 12% extra' also listed below. A 'Get Quote' button is located to the left of a 'Message' dialog box. The 'Message' box has an 'i' icon and the text 'Please enter a number for the length, letters and characters are not allowed'. It has an 'OK' button. To the right of the message box is a 'Refresh' button and a 'Product Cost:' section with a blank box. At the bottom are 'Remove Pipe' and 'Remove all pipes' buttons, and a 'Get receipt' button. The status bar at the bottom shows 'Order 1 cost: 0.00'.

You can see that this type of input is not accepted and the user is shown the error message detailing the problem.

This test has passed.

Outer diameter check:

**Long Pipes**

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade:	Colour print:	Inner insulation:	Outer reinforcement:	Chemical resistance:
1	No colours	No	No	No

Length: (M) Outer Diameter: (Inches) Quantity:

3.2	2.klk	1
-----	-------	---

Extras pricing:

1 Colour - 12% extra
2 Colours - 16% extra

[Get Quote](#)

Product details for order 1:  
No products have been ordered at the moment, please select the products details to continue

[Remove Pipe](#) [Remove all pipes](#)

Order 1 cost: 0.00

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

[fresh](#) [Product Cost:](#)

[Get receipt](#)

**Message**

Please enter a number for the outer diameter, letters and characters are not allowed

[OK](#)

You can see that this type of input is not accepted and the user is shown the error message detailing the problem.

This test has passed.

All of the subtests from this part of the Error Checking for type errors are complete. We will now move on to range errors.

Range errors:

Range errors occur when the user inputs a value that is either above or below our accepted range. For example, if our accepted range is 1-10 and the user inputs -1, an error should be displayed detailing this problem. This is the same if the user attempts to input 14 as this value is out of bounds of our range and an error message should therefore be shown.

Length check.

When the user attempts to input negative value into the length field, an error message should be shown.

Long Pipes - Order form

## Long Pipes

Create Order   Remove all orders   Remove Order (current)

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

Length: (M) Outer Diameter: (Inches) Quantity:

1   No colours   No   No   No

-9   2   1

Extras pricing:  
1 Colour - 12% extra  
2 Colours - 16% extra

Inner insulation - 13% extra

Get Quote   Message   Refresh   Product Cost:   Get receipt

No products have been ordered at the moment, please select the products details to continue

Please enter a valid length, the length must greater than 0 and no greater than 6M

OK

Remove Pipe   Remove all pipes   Order 1 cost: 0.00

The user enters -9 and the appropriate error message is displayed for this.

This test has passed.

Now we will check if the user enters a value beyond the accepted range.

## Long Pipes

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

Length: (M) Outer Diameter: (Inches) Quantity: Extras pricing:

12 2 1 1 Colour - 12% extra

[Get Quote](#) [Refresh](#) [Product Cost:](#)

Product details for order 1:  
No products have been ordered at the moment, please select the products details to continue

[Remove Pipe](#) [Remove all pipes](#) [Get receipt](#)

Order 1 cost: 0.00

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

**Additional options:**

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

**Message**

Please enter a valid length, the length must greater than 0 and no greater than 6M

**OK**

This is what we expect to see and length is behaving properly.

This test has passed.

Outer diameter:

The screenshot shows a Windows application window titled "Long Pipes". The main interface includes buttons for "Create Order", "Remove all orders", and "Remove Order (current)". A dropdown menu shows "Order: 1". Below these are sections for "Select your pipe options" and "How to use:". The "Select your pipe options" section contains dropdown menus for Plastic grade (1), Colour print (No colours), Inner insulation (Yes), Outer reinforcement (Yes), and Chemical resistance (Yes). It also includes fields for Length (M) (2), Outer Diameter (Inches) (-3), and Quantity (2). An "Extras pricing:" section lists additional costs: 1 Colour - 12% extra, 2 Colours - 16% extra, and Inner insulation - 13% extra. A "Get Quote" button is present. A "Message" dialog box is overlaid on the form, displaying an error message: "Please enter a valid outer diameter, the outer diameter must greater than 0 and no greater than 2 Inches". The "OK" button of the dialog is visible. At the bottom, there are buttons for "Remove Pipe", "Remove all pipes", and "Get receipt". The status bar at the bottom shows "Order 1 subtotal: £ 0.00".

Here the user types in -3, as this is not in our accepted range an appropriate error message has been displayed.

This test has passed.

Next we will check that if the user enter a value above the accepted range, an appropriate error message is shown.

## Long Pipes

Order: 1

Select your pipe options:

Plastic grade:	Colour print:	Inner insulation:	Outer reinforcement:	Chemical resistance:
<input type="button" value="1"/>	<input type="button" value="No colours"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>

Length: (M) Outer Diameter: (Inches) Quantity: Extras pricing:

<input type="button" value="2"/>	<input type="button" value="5"/>	<input type="button" value="2"/>	1 Colour - 12% extra 2 Colours - 16% extra Inner insulation - 13% extra
----------------------------------	----------------------------------	----------------------------------	---

Message  
i Please enter a valid outer diameter, the outer diameter must greater than 0 and no greater than 2 Inches

Product details for

No products have been added to the order.

Order 1 subtotal: £ 0.00

The user has entered a value that is above the accepted range and an appropriate error message has been displayed.

This test has passed.

Next we will check that the quantity range checks are working.

Quantity:

When the user enters a -4, the program should respond with an appropriate error message.

Long Pipes - Order Form

## Long Pipes

Create Order   Remove all orders   Remove Order (current)

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

Length: (M) Outer Diameter: (Inches) Quantity:

1   No colours   Yes   Yes   Yes

2   2   -2

Extras pricing:

1 Colour - 12% extra  
2 Colours - 16% extra

Get Quote   Message

Please enter a valid quantity, the quantity must be greater than 0 and no greater than 100

OK

Product details for order 1

No products have been ordered at the moment, please select the products details to continue

Remove Pipe   Remove all pipes   Get receipt

Order 1 subtotal: £ 0.00

How to use:

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

Refresh   Product Cost:

The appropriate error message has now been displayed to the user indicating that -2 is not allowed.

This test has passed.

We now need to check that an appropriate error message is displayed when the user enters a value beyond the accepted range.

**Long Pipes**

**Create Order**   **Remove all orders**   **Remove Order (current)**

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

Length: (M) Outer Diameter: (Inches) Quantity: Extras pricing:

1 No colours Yes Yes Yes  
2 2 150 1 Colour - 12% extra  
2 Colours - 16% extra

**Get Quote**

**Message**

Please enter a valid quantity, the quantity must be greater than 0 and no greater than 100

**OK**

Product details for order

No products have been ordered at the moment, please select the products details to continue

**Remove Pipe**   **Remove all pipes**   **Get receipt**

Order 1 subtotal: £ 0.00

How to use:  
1. Create an order  
2. Select your desired pipe options  
3. Click get quote  
4. When finished, click get receipt  
Additional options:  
1. Remove an order - Deletes the entire order  
2. Remove pipe - Deletes the selected pipe

The appropriate error message has now been displayed to the user.

This test has passed.

All range error checks are complete.

Test 3 - Appropriate error checking at all levels are handled appropriately when adding a pipe – Summary

12/12 passes so this test has passed.

Test 4 – Removing multiple pipes

When a user has made an order and has added multiple pipes, we will now check to see if remove pipe and remove multiple pipes buttons function as expected.

Remove a pipe button:

When the user does not select a pipe, an appropriate error message should be shown.

## Long Pipes

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade: 1 Colour print: No colours Inner insulation: Yes Outer reinforcement: Yes Chemical resistance: Yes

Length: (M) Outer Dia:

[Get Quote](#)

Message

No pipe selected - please select a pipe to continue.

OK

Chemical resistance - 14% extra

Product details for order 1:

Pipe grade: 1, Length: 2.000, Outer diameter: 2.0, Quantity: 1, Chemical resistance: No, Colours: No  
Pipe grade: 1, Length: 1.000, Outer diameter: 1.0, Quantity: 1, Chemical resistance: No, Colours: No

[Remove Pipe](#) [Remove all pipes](#) [Get receipt](#)

Order 1 cost: 21.15

Refresh

Product Cost:  
18.80  
2.35

How to use:

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

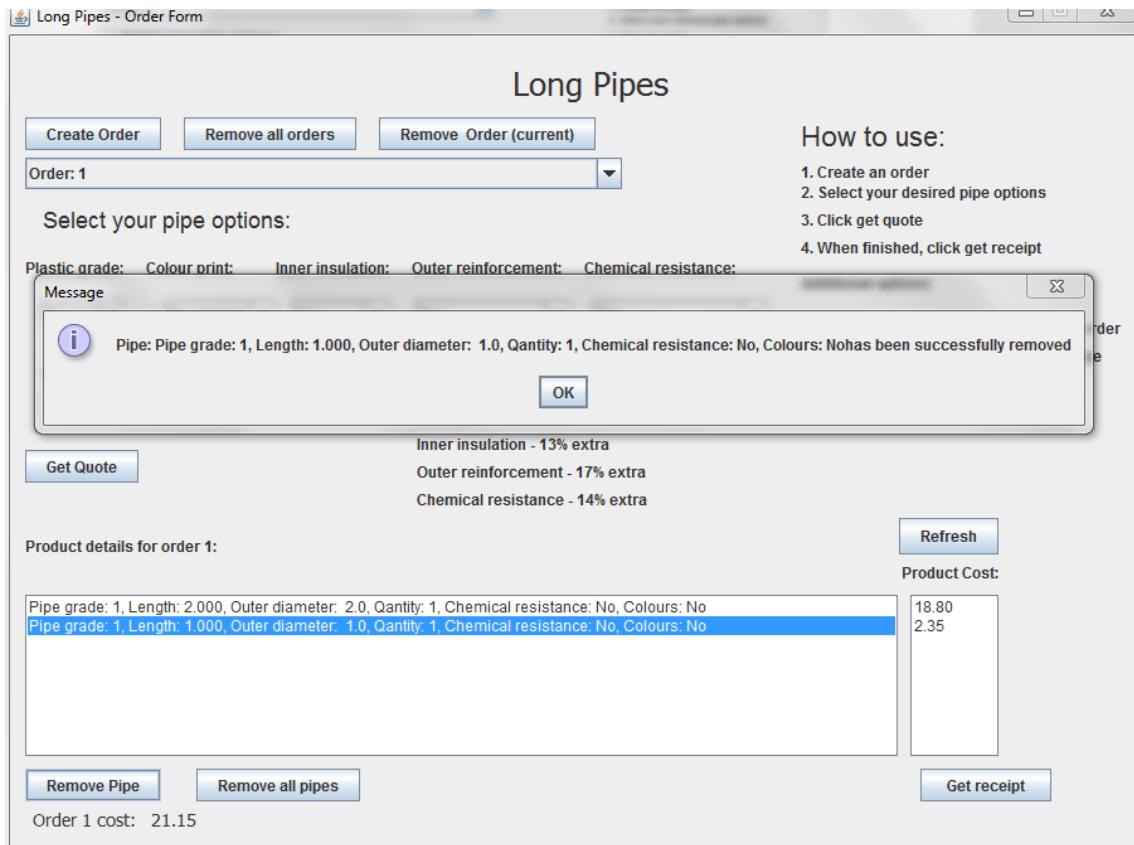
Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

The remove Pipe button responds with correct error message.

This test has passed.

Next we shall check that when a user selects a pipe to be remove, it is removed as expected.



The program responds by displaying the correct message.

This test has passed.

After the user clicks OK.

Long Pipes - Order Form

## Long Pipes

Create Order   Remove all orders   Remove Order (current)

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1   No colours   Yes   Yes   Yes

Length: (M) Outer Diameter: (Inches) Quantity:

Extras pricing:

1 Colour - 12% extra  
2 Colours - 16% extra  
Inner insulation - 13% extra  
Outer reinforcement - 17% extra  
Chemical resistance - 14% extra

Get Quote   Refresh

Product details for order 1:

Pipe grade: 1, Length: 2.000, Outer diameter: 2.0, Quantity: 1, Chemical resistance: No, Colours: No

Product Cost: 18.80

Remove Pipe   Remove all pipes   Get receipt

Order 1 cost: 18.80

This is the correct behavior of the remove pipe button.

This test has passed.

We shall now continue.

Next we will see that when the user wishes to remove all products from an order. The program responds accordingly and all pipes from the selected order are removed.

Before remove all pipes button is clicked:

## Long Pipes

[Create Order](#)
[Remove all orders](#)
[Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade:  Colour print:  Inner insulation:  Outer reinforcement:  Chemical resistance:

Length: (M)	Outer Diameter: (Inches)	Quantity:	<b>Extras pricing:</b>
<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	1 Colour - 12% extra 2 Colours - 16% extra Inner insulation - 13% extra Outer reinforcement - 17% extra Chemical resistance - 14% extra

[Get Quote](#)

Product details for order 1:

Pipe grade: 1, Length: 2.000, Outer diameter: 2.0, Quantity: 1, Chemical resistance: No, Colours: No  
 Pipe grade: 1, Length: 1.000, Outer diameter: 1.0, Quantity: 1, Chemical resistance: No, Colours: No  
 Pipe grade: 3, Length: 1.000, Outer diameter: 1.0, Quantity: 11, Chemical resistance: No, Colours: No

[Remove Pipe](#) [Remove all pipes](#)

Order 1 cost: 69.66

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

[Refresh](#)

**Product Cost:**

18.80
2.35
48.51

[Get receipt](#)

After remove pipes button has been clicked:

## Long Pipes

[Create Order](#)   [Remove all orders](#)   [Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1	No colours	Yes	Yes	Yes
---	------------	-----	-----	-----

Length: (M)   Outer Diameter: (Inches)   Quantity:

--	--	--

[Get Quote](#)

Product details for order 1:

```
Pipe grade: 1, Length: 2.000, Outer diameter: 2.0, Qantity: 1, Chemical resistance: No, Colours: No
Pipe grade: 1, Length: 1.000, Outer diameter: 1.0, Qantity: 1, Chemical resistance: No, Colours: No
Pipe grade: 3, Length: 1.000, Outer diameter: 1.0, Qantity: 11, Chemical resistance: No, Colours: No
```

[Remove Pipe](#)   [Remove all pipes](#)

Order 1 cost: 69.66

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

[Refresh](#)

Product Cost:

18.80
2.35
48.51

[Get receipt](#)

Message

All pipes have been successfully removed.

[OK](#)

When the user clicks OK, the products from order should be deleted.

This test has passed.

After OK is clicked:

## Long Pipes

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1 ▾ No colours ▾ Yes ▾ Yes ▾ Yes ▾

Length: (M) Outer Diameter: (Inches) Quantity:

Get Quote

Extras pricing:

1 Colour - 12% extra  
2 Colours - 16% extra  
Inner insulation - 13% extra  
Outer reinforcement - 17% extra  
Chemical resistance - 14% extra

Product details for order 1:

No products have been ordered at the moment, please select the products details to continue

Refresh

Product Cost:

Get receipt

Remove Pipe Remove all pipes

Order 1 cost: 0.00

The remove pipes button works as expected.

This test has passed.

This test passes.

5/5 passes so this test has PASSED.

### Test 5 – Multiple order selection

Here we will test that the following change when the user change which order they wish to see i.e. its contents.

**Long Pipes**

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

**Additional options:**

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

**Select your pipe options:**

Plastic grade: 1 Colour print: No colours Inner insulation: Yes Outer reinforcement: Yes Chemical resistance: Yes

Length: (M) Outer Diameter: (Inches) Quantity:

**Extras pricing:**

- 1 Colour - 12% extra
- 2 Colours - 16% extra
- Inner insulation - 13% extra
- Outer reinforcement - 17% extra
- Chemical resistance - 14% extra

**Product details for order 1:**

Pipe grade: 5, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: No, Colours: 2, Insulated: Yes, Reinforced: Yes

**Product Cost:** 8.15

**Get Quote**      **Refresh**

**Remove Pipe**      **Remove all pipes**

**Order 1 subtotal: £ 8.15**

**Get receipt**

1. We need to see that the order selected changes to the newly selected order
2. We need to check that this label changes to the newly selected order
3. We need to check that the products in the newly selected order are displayed.
4. We need to check that the order subtotal values change and that the label changes to the newly selected order

Order 2 selected:

**Long Pipes**

**Create Order      Remove all orders      Remove Order (current)**

**Order: 2**

Select your pipe options: 1

Plastic grade:  Colour print:  Inner insulation:  Outer reinforcement:  Chemical resistance:

Length: (M)  Outer Diameter: (Inches)  Quantity:

**Extras pricing:**

1 Colour - 12% extra  
2 Colours - 16% extra  
Inner insulation - 13% extra  
Outer reinforcement - 17% extra  
Chemical resistance - 14% extra

**Get Quote**

**Product details for order 2:** 2

Pipe grade: 3, Length: 1.000, Outer diameter: 1.0, Quantity: 11, Chemical resistance: Yes, Colours: No

**Refresh**

**Product Cost:** 3

**Remove Pipe** **Remove all pipes** 4

Order 2 subtotal: £  4

**Get receipt**

We can see that these changes were expected and areas 1, 2, 3, 4 did change as expected. These tests has passed.

Test 5 – Multiple order selection summary:

1/1 passes so this test has passed.

Test 6 – Calculation of an order

## Long Pipes

[Create Order](#)   [Remove all orders](#)   [Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade:	Colour print:	Inner insulation:	Outer reinforcement:	Chemical resistance:
<input type="button" value="1"/>	<input type="button" value="No colours"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>

Length: (M)   Outer Diameter: (Inches)   Quantity:

[Get Quote](#)

Product details for order 1:

Pipe grade: 5, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: No, Colours: 2, Insulated: Yes, Reinforced: Yes

[Remove Pipe](#)   [Remove all pipes](#)

Order 1 subtotal: £ 8.15

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

[Refresh](#)

Product Cost:

[Get receipt](#)

Here we have an order that has 1 product and the price for this pipe is £8.15.

If we the user adds a new product, the total should be calculated correctly.

## Long Pipes

[Create Order](#)

[Remove all orders](#)

[Remove Order \(current\)](#)

Order: 1

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

Select your pipe options:

Plastic grade:	Colour print:	Inner insulation:	Outer reinforcement:	Chemical resistance:
<input type="button" value="1"/>	<input type="button" value="No colours"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>

Length: (M)   Outer Diameter: (Inches)   Quantity:

<input type="text"/>	<input type="text"/>	<input type="text"/>
----------------------	----------------------	----------------------

**Extras pricing:**

- 1 Colour - 12% extra
- 2 Colours - 16% extra
- Inner insulation - 13% extra
- Outer reinforcement - 17% extra
- Chemical resistance - 14% extra

**Get Quote**

Product details for order 1:

Pipe grade: 5, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: No, Colours: 2, Insulated: Yes, Reinforced: Yes  
 Pipe grade: 4, Length: 2.00, Outer diameter: 2.0, Quantity: 1, Chemical resistance: No, Colours: 2, Insulated: Yes, Reinforced: Yes

**Remove Pipe**

**Remove all pipes**

Order 1 cost: 63.05

**Refresh**

**Product Cost:**

8.15
54.90

We can see that the next products' value is £54.90. In the subtotal below (2) we can see that the value now attained is £63.05. This value is correct as  $8.15 + 54.90 = 63.05$ . The calculation has been correct as what we would have expected.

This subtest has passed.

We shall now add multiple products and see if the calculation is correct.

## Long Pipes

[Create Order](#)

[Remove all orders](#)

[Remove Order \(current\)](#)

Order: 1

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

**Additional options:**

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

**Select your pipe options:**

Plastic grade:	Colour print:	Inner insulation:	Outer reinforcement:	Chemical resistance:
<input type="button" value="1"/>	<input type="button" value="No colours"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>

Length: (M)   Outer Diameter: (Inches)   Quantity:

<input type="text"/>	<input type="text"/>	<input type="text"/>
----------------------	----------------------	----------------------

**Get Quote**

**Extras pricing:**

- 1 Colour - 12% extra
- 2 Colours - 16% extra
- Inner insulation - 13% extra
- Outer reinforcement - 17% extra
- Chemical resistance - 14% extra

**Product details for order 1:**

```
Pipe grade: 5, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: No, Colours: 2, Insulated: Yes, Reinforced: Yes
Pipe grade: 4, Length: 2.00, Outer diameter: 2.0, Quantity: 1, Chemical resistance: No, Colours: 2, Insulated: Yes, Reinforced: Yes
Pipe grade: 3, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: Yes, Colours: 2
Pipe grade: 1, Length: 1.000, Outer diameter: 1.0, Quantity: 67, Chemical resistance: No, Colours: No
```

**Order 1 cost:** 226.33

**Product Cost:**

8.15
54.90
5.83
157.45

**Get receipt**

We can see that the user has entered 4 products and their values are:

1. 8.15
2. 54.90
3. 5.83
4. 157.45

This calculates to:  $8.15 + 54.90 + 5.83 + 157.45 = 226.33$ . This is the correct value that has been calculated.

This subtest has passed.

### Test 6 – Calculation of an order – Summary

2/2 passes so this test has passed.

### Test 7 – Calculation and update of multiple orders

In this test we will check that the addition of multiple orders has been added correctly.

Order 1 details:

**Long Pipes**

**Create Order**   **Remove all orders**   **Remove Order (current)**

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1   No colours   Yes   Yes   Yes

Length: (M)   Outer Diameter: (Inches)   Quantity:

Get Quote

Extras pricing:

1 Colour - 12% extra  
2 Colours - 16% extra  
Inner insulation - 13% extra  
Outer reinforcement - 17% extra  
Chemical resistance - 14% extra

Product details for order 1:

Pipe grade: 1, Length: 1.000, Outer diameter: 1.0, Quantity: 1, Chemical resistance: Yes, Colours: No

Product Cost: 2.68

Refresh

Remove Pipe   Remove all pipes   Get receipt

Order 1 cost: 2.68

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

The value of order 1 is £2.68.

Order 2 details:

### Long Pipes

[Create Order](#)
[Remove all orders](#)
[Remove Order \(current\)](#)

Order: 2

Select your pipe options:

Plastic grade:  Colour print:  Inner insulation:  Outer reinforcement:  Chemical resistance:

Length: (M)  Outer Diameter: (Inches)  Quantity:

[Get Quote](#)

Product details for order 2:

Pipe grade: 3, Length: 1.000, Outer diameter: 1.0, Quantity: 11, Chemical resistance: Yes, Colours: No

Order 2 subtotal: £ 55.30

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

[Refresh](#)

**Product Cost:**

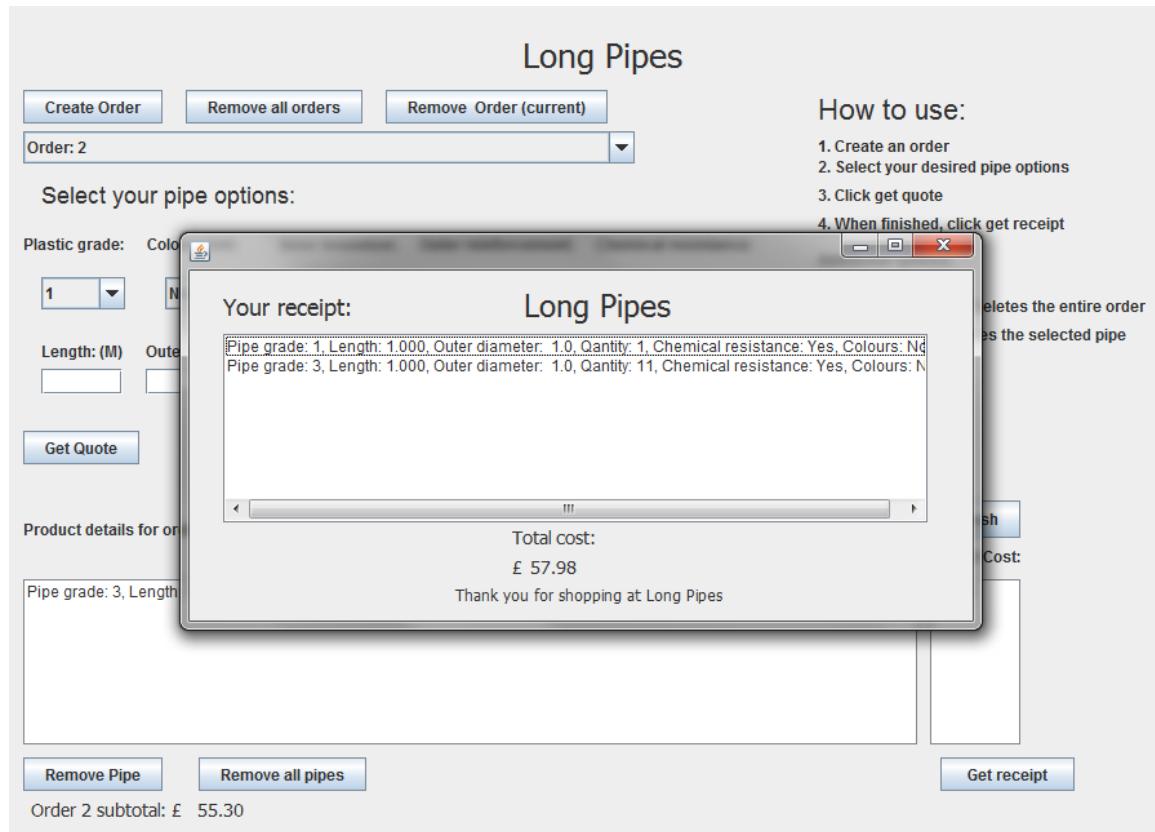
55.30
-------

[Get receipt](#)

Order 2 subtotal is £55.30.

When we check the receipt, the value that we should receive is £55.30+£2.68 = £57.97.  
We will now check.

Receipt screen total:

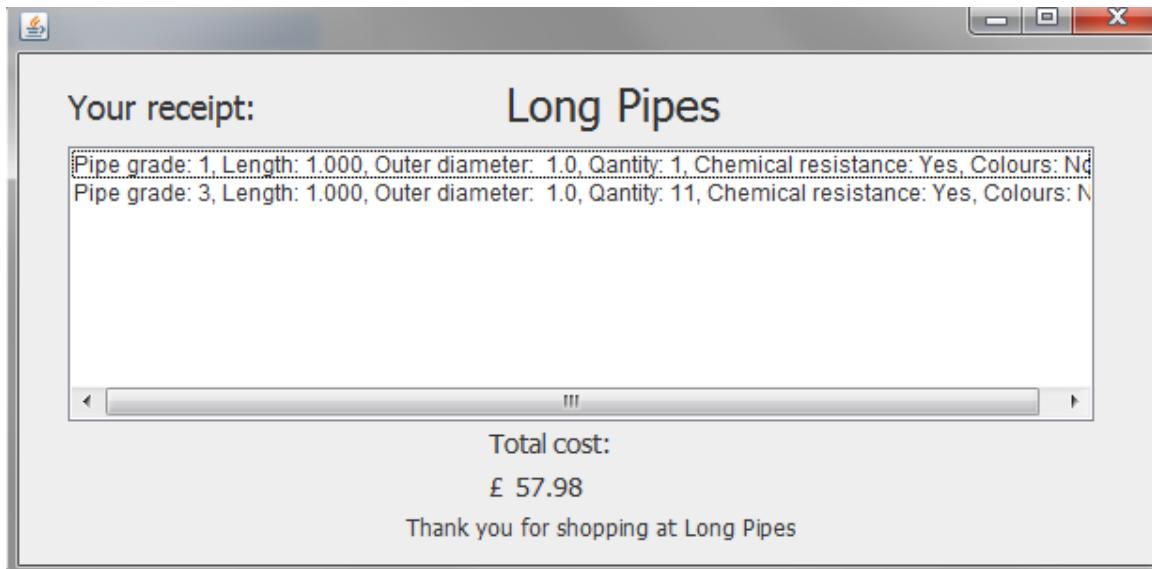


Here we see that this is the correct value that we attained.

This subtest has passed.

Next we will check that when the user removes a product, the receipt screen updates accordingly.

Current state of receipt screen.



Current state of Order 1:

**Long Pipes**

Create Order   Remove all orders   Remove Order (current)

Order: 1

Select your pipe options:

Plastic grade: 1   Colour print: No colours   Inner insulation: Yes   Outer reinforcement: Yes   Chemical resistance: Yes

Length: (M) 1   Outer Diameter: (Inches) 1.0   Quantity: 1

Get Quote

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

Product details for order 1:

Pipe grade: 1, Length: 1.000, Outer diameter: 1.0, Quantity: 1, Chemical resistance: Yes, Colours: No	Product Cost: 2.68
---	--------------------

Remove Pipe   Remove all pipes   Get receipt

Order 1 subtotal: £ 2.68

Current state of order 2:

**Long Pipes**

**Create Order**   **Remove all orders**   **Remove Order (current)**

Order: 2

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1   No colours   Yes   Yes   Yes

Length: (M)   Outer Diameter: (Inches)   Quantity:

1 Colour - 12% extra  
2 Colours - 16% extra  
Inner insulation - 13% extra  
Outer reinforcement - 17% extra  
Chemical resistance - 14% extra

**Get Quote**

Product details for order 2:

Pipe grade: 3, Length: 1.000, Outer diameter: 1.0, Qantity: 11, Chemical resistance: Yes, Colours: No

**Refresh**

Product Cost: 55.30

**Remove Pipe**   **Remove all pipes**   **Get receipt**

Order 2 subtotal: £ 55.30

Removing product 1 from order 2:

**Long Pipes**

**Create Order**   **Remove all orders**   **Remove Order (current)**

Order: 2

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1   No colours   Yes   Yes   Yes

**Message**

**i** Pipe: Pipe grade: 3, Length: 1.000, Outer diameter: 1.0, Quantity: 11, Chemical resistance: Yes, Colours: Nohas been successfully removed

**OK**

Product details for order 2:

Pipe grade: 3, Length: 1.000, Outer diameter: 1.0, Quantity: 11, Chemical resistance: Yes, Colours: No

Product Cost: 55.30

**Remove Pipe**   **Remove all pipes**   **Get receipt**

Order 2 subtotal: £ 55.30

After user clicks OK, the state of order 2:

**Long Pipes**

**Create Order**   **Remove all orders**   **Remove Order (current)**

Order: 2

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1   No colours   Yes   Yes   Yes

Length: (M)   Outer Diameter: (Inches)   Quantity:

Get Quote

Extras pricing:

1 Colour - 12% extra  
2 Colours - 16% extra  
Inner insulation - 13% extra  
Outer reinforcement - 17% extra  
Chemical resistance - 14% extra

Product details for order 2:

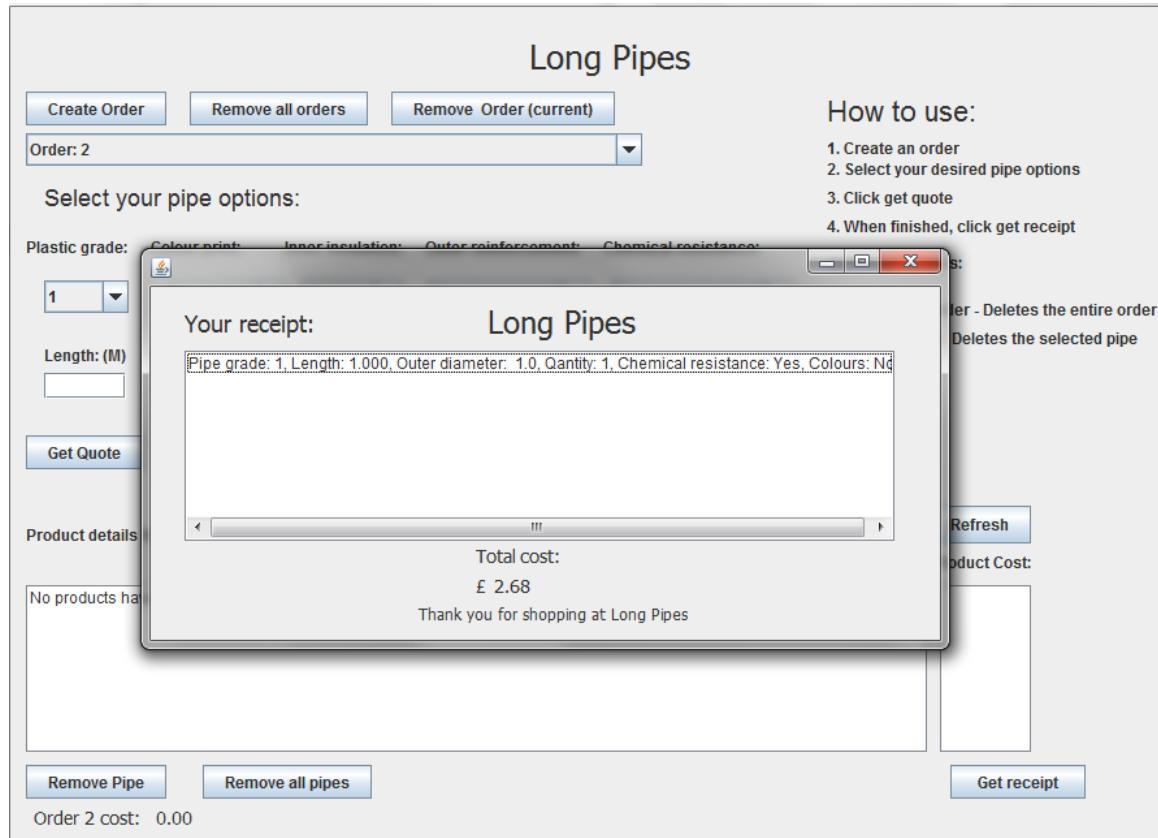
No products have been ordered at the moment, please select the products details to continue

Product Cost:

**Remove Pipe**   **Remove all pipes**   **Get receipt**

Order 2 cost: 0.00

The new state of the receipt screen:



We can see that as we have now removed product 1 from order 2. Only order 1's product is shown now as expected.

This subtest has passed.

Now we have completed testing the update and calculating of multiple orders.

Test 7 – Calculation and update of multiple orders – Summary

2/2 passes so this test has PASSED.

Test 8 – Receipt screen information

When the user enters a pipe of grade 1, no colors, no insulation, no outer reinforcement and no chemical resistance. This should show on the product screen as well as entering an outer diameter of 2, quantity of 50 and a length of 5M.

Ordering a pipe:

**Long Pipes**

**Create Order**   **Remove all orders**   **Remove Order (current)**

**Order: 1**

**Select your pipe options:**

Plastic grade:  Colour print:  Inner insulation:  Outer reinforcement:  Chemical resistance:

Length: (M)  Outer Diameter: (Inches)  Quantity:

**Extras pricing:**

1 Colour - 12% extra  
2 Colours - 16% extra

**Get Quote**

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

**Additional options:**

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

**Product details for order 1:**

No products have been ordered at the moment, please select the products details to continue

**Remove Pipe**   **Remove all pipes**   **Get receipt**

Order 1 subtotal: £ 0.00

**Message**

Pipe has been sucessfully added, please continue.

**OK**

After the user clicks OK, these details should now be shown on-screen in the product list.

After the user clicks OK:

The screenshot shows a software interface for configuring 'Long Pipes'. At the top, there are three buttons: 'Create Order', 'Remove all orders', and 'Remove Order (current)'. Below them is a dropdown menu set to 'Order: 1'. On the left, there's a section titled 'Select your pipe options:' with dropdown menus for Plastic grade (set to 1), Colour print (No colours), Inner insulation (Yes), Outer reinforcement (Yes), and Chemical resistance (Yes). To the right of this is a 'How to use:' section with four numbered steps: 1. Create an order, 2. Select your desired pipe options, 3. Click get quote, 4. When finished, click get receipt. Below these are 'Additional options' for removing an order or pipe. In the center, there's a 'Extras pricing:' section listing additional costs for color, insulation, reinforcement, and chemical resistance. On the far right, there's a 'Product Cost' section showing a cost of 2350.00. At the bottom, there are buttons for 'Get Quote', 'Refresh', 'Get receipt', 'Remove Pipe', and 'Remove all pipes'. The total cost for Order 1 is displayed as 2350.00.

As shown above, the correct details have been shown for this product.

This test has passed.

We will check another product see that it does in fact show in the product list.

This time we require a pipe with grade 4, no chemical resistance, and 2 colors and with inner insulation and outer reinforcement.

Quantity – 30

Length – 6M

Outer diameter – 2Inches

Before add quote is clicked:

**Long Pipes**

**Create Order**
**Remove all orders**
**Remove Order (current)**

Order: 1
 ▼

**Select your pipe options:**

Plastic grade:
Colour print:
Inner insulation:
Outer reinforcement:
Chemical resistance:

4
2 colours
Yes
Yes
No

Length: (M)
Outer Diameter: (Inches)
Quantity:

6
2
30

**Extras pricing:**

1 Colour - 12% extra
2 Colours - 16% extra
Inner insulation - 13% extra
Outer reinforcement - 17% extra
Chemical resistance - 14% extra

**Get Quote**
**Refresh**

**Product details for order 1:**

Pipe grade: 1, Length: 5.000, Outer diameter: 2.0, Quantity: 50, Chemical resistance: No, Colours: No

Product Cost:

2350.00

**Remove Pipe**
**Remove all pipes**
**Get receipt**

Order 1 cost: 2350.00

After add quote is clicked.

**Long Pipes**

**Create Order**   **Remove all orders**   **Remove Order (current)**

Order: 1

Select your pipe options:

Plastic grade: 1 Colour print: No colours Inner insulation: Yes Outer reinforcement: Yes Chemical resistance: Yes

Length: (M) Outer Diameter: (Inches) Quantity: 1

**Get Quote**

Product details for order 1:

Pipe grade: 1, Length: 5.000, Outer diameter: 2.0, Quantity: 50, Chemical resistance: No, Colours: No	2350.00
Pipe grade: 4, Length: 6.00, Outer diameter: 2.0, Quantity: 30, Chemical resistance: No, Colours: 2, Insulated: Yes, Reinforced: Yes	4940.70

**Remove Pipe**   **Remove all pipes**

Order 1 cost: 7290.70

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

**Additional options:**

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

**Refresh**

**Product Cost**

**Get receipt**

We can see that the correct information has been shown as we would expect.

This test has passed.

#### Test 8 –Receipt screen information – Summary

2/2 passes so this test has PASSED.

Testing Results:

ID:	Individual test results::	Overall result:
1	8/8 PASSES	PASSED
2	7/7 PASSES	PASSED
3	7/7 PASSES	PASSED
4	12/12 PASSES	PASSED
5	1/1 PASSES	PASSED
6	2/2 PASSES	PASSED
7	2/2 PASSS	PASSED
8	2/2 PASSES	PASSED

All of our test areas have passed.

## Sample usage

Here we will demonstrate the sample usage of this application.

Starting the application:

The screenshot shows the 'Long Pipes' application interface. At the top, there are three buttons: 'Create Order', 'Remove all orders', and 'Remove Order (current)'. Below these, a section titled 'Select your pipe options:' contains dropdown menus for Plastic grade (set to 1), Colour print (No colours), Inner insulation (Yes), Outer reinforcement (Yes), and Chemical resistance (Yes). To the right, a 'How to use:' section lists steps 1-4: Create an order, Select your desired pipe options, Click get quote, and When finished, click get receipt. Below this, an 'Extras pricing:' table shows additional costs for 1 Colour (12% extra), 2 Colours (16% extra), Inner insulation (13% extra), Outer reinforcement (17% extra), and Chemical resistance (14% extra). A note at the bottom left says 'Unable to show product details for an order at the moment, consider making an order.' On the right, there are 'Refresh', 'Product Cost', and 'Get receipt' buttons. At the bottom, there are 'Remove Pipe' and 'Remove all pipes' buttons, and a note 'Unable to calculate order subtotal.'

Upon start we see the screen above. Next we will create an order and add some products.

The screenshot shows the 'Long Pipes' application interface after creating an order. The 'Order: 1' dropdown now shows 'Order: 1'. A confirmation message box is displayed in the center: 'Message' (info icon) Successfully created a new order, please proceed. An 'OK' button is at the bottom. The rest of the interface is identical to the first screenshot, including the 'How to use:' section, extras pricing table, and bottom buttons.

Now we will add products.

### Long Pipes

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1 No colours Yes Yes Yes

Length: (M) Outer Diameter: (Inches) Quantity:

Get Quote

How to use:

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

Extras pricing:

- 1 Colour - 12% extra
- 2 Colours - 16% extra
- Inner insulation - 13% extra
- Outer reinforcement - 17% extra
- Chemical resistance - 14% extra

Product details for order 1:

Pipe grade: 3, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: Yes, Colours: 1  
 Pipe grade: 1, Length: 5.000, Outer diameter: 2.0, Quantity: 5, Chemical resistance: Yes, Colours: No

Refresh

Product Cost:  
5.63  
267.90

Get receipt

Remove Pipe Remove all pipes

Order 1 cost: 273.53

Next we will create 2 new order.

### Long Pipes

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1 No colours Yes Yes Yes

Length: (M) Outer Diameter: (Inches) Quantity:

Get Quote

How to use:

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

Message

Successfully created a new order, please proceed

OK

Extras pricing:

- 1 Colour - 12% extra
- 2 Colours - 16% extra
- Inner insulation - 13% extra
- Outer reinforcement - 17% extra
- Chemical resistance - 14% extra

Product details for order 1:

Pipe grade: 3, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: Yes, Colours: 1  
 Pipe grade: 1, Length: 5.000, Outer diameter: 2.0, Quantity: 5, Chemical resistance: Yes, Colours: No

Refresh

Product Cost:  
5.63  
267.90

Get receipt

Remove Pipe Remove all pipes

Order 1 cost: 273.53

Order 2:

**Long Pipes**

<input type="button" value="Create Order"/> <input type="button" value="Remove all orders"/> <input type="button" value="Remove Order (current)"/>		
Order: 2		
<b>Select your pipe options:</b> Plastic grade: <input type="button" value="Colour print"/> Inner insulation: <input type="button" value="Yes"/> Outer reinforcement: <input type="button" value="Yes"/> Chemical resistance: <input type="button" value="Yes"/>  Length: (M) <input type="button" value="Outer Diameter: (Inches)"/> Quantity: <input type="button" value=""/>		
<b>How to use:</b> 1. Create an order 2. Select your desired pipe options 3. Click get quote 4. When finished, click get receipt  <b>Additional options:</b> 1. Remove an order - Deletes the entire order 2. Remove pipe - Deletes the selected pipe		
<b>Extras pricing:</b> 1 Colour - 12% extra 2 Colours - 16% extra Inner insulation - 13% extra Outer reinforcement - 17% extra Chemical resistance - 14% extra		
<b>Product details for order 2:</b> No products have been ordered at the moment, please select the products details to continue		
<input type="button" value="Get Quote"/> <input type="button" value="Refresh"/> <input type="button" value="Product Cost: No items"/>		
<input type="button" value="Remove Pipe"/> <input type="button" value="Remove all pipes"/> <input type="button" value="Get receipt"/>		
Order 2 subtotal: £ 0.00		

Order 3:

**Long Pipes**

<input type="button" value="Create Order"/> <input type="button" value="Remove all orders"/> <input type="button" value="Remove Order (current)"/>		
Order: 3		
<b>Select your pipe options:</b> Plastic grade: <input type="button" value="Colour print"/> Inner insulation: <input type="button" value="Yes"/> Outer reinforcement: <input type="button" value="Yes"/> Chemical resistance: <input type="button" value="Yes"/>  Length: (M) <input type="button" value="Outer Diameter: (Inches)"/> Quantity: <input type="button" value=""/>		
<b>How to use:</b> 1. Create an order 2. Select your desired pipe options 3. Click get quote 4. When finished, click get receipt  <b>Additional options:</b> 1. Remove an order - Deletes the entire order 2. Remove pipe - Deletes the selected pipe		
<b>Extras pricing:</b> 1 Colour - 12% extra 2 Colours - 16% extra Inner insulation - 13% extra Outer reinforcement - 17% extra Chemical resistance - 14% extra		
<b>Product details for order 3:</b> No products have been ordered at the moment, please select the products details to continue		
<input type="button" value="Get Quote"/> <input type="button" value="Refresh"/> <input type="button" value="Product Cost: No items"/>		
<input type="button" value="Remove Pipe"/> <input type="button" value="Remove all pipes"/> <input type="button" value="Get receipt"/>		
Order 3 subtotal: £ 0.00		

Next we will add products to both orders and require a receipt.

Order 1 details:

**Long Pipes**

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 1

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

Length: (M) Outer Diameter: (Inches) Quantity:

Get Quote

How to use:

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

Extras pricing:

- 1 Colour - 12% extra
- 2 Colours - 16% extra
- Inner insulation - 13% extra
- Outer reinforcement - 17% extra
- Chemical resistance - 14% extra

Product details for order 1:

Pipe grade: 3, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: Yes, Colours: 1  
 Pipe grade: 1, Length: 5.000, Outer diameter: 2.0, Quantity: 5, Chemical resistance: Yes, Colours: No

Product Cost: 5.63 267.90

Refresh

Get receipt

Remove Pipe Remove all pipes

Order 1 subtotal: £ 273.53

Order 2 details:

**Long Pipes**

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 2

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

Length: (M) Outer Diameter: (Inches) Quantity:

Get Quote

How to use:

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

Extras pricing:

- 1 Colour - 12% extra
- 2 Colours - 16% extra
- Inner insulation - 13% extra
- Outer reinforcement - 17% extra
- Chemical resistance - 14% extra

Product details for order 2:

Pipe grade: 1, Length: 6.000, Outer diameter: 2.0, Quantity: 33, Chemical resistance: Yes, Colours: No  
 Pipe grade: 5, Length: 2.00, Outer diameter: 2.0, Quantity: 79, Chemical resistance: Yes, Colours: 2, Insulated: Yes, Reinforced: Yes

Product Cost: 2121.77 5871.01

Refresh

Get receipt

Remove Pipe Remove all pipes

Order 2 subtotal: £ 7992.78

Order 3 details:

### Long Pipes

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 3

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1 No colours Yes Yes Yes

Length: (M) Outer Diameter: (Inches) Quantity:

1000 200 100

[Get Quote](#)

How to use:

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

Extras pricing:

1 Colour - 12% extra  
 2 Colours - 16% extra  
 Inner insulation - 13% extra  
 Outer reinforcement - 17% extra  
 Chemical resistance - 14% extra

Product details for order 3:

Pipe grade: 4, Length: 6.00, Outer diameter: 2.0, Qantity: 90, Chemical resistance: Yes, Colours: 2, Insulated: Yes, Reinforced: Yes

[Refresh](#)

Product Cost: 16897.19

[Remove Pipe](#) [Remove all pipes](#) [Get receipt](#)

Order 3 cost: 16897.19

Now I will request a receipt.

### Long Pipes

Your order details:

This is the products that you have ordered from all of your order that you have made.

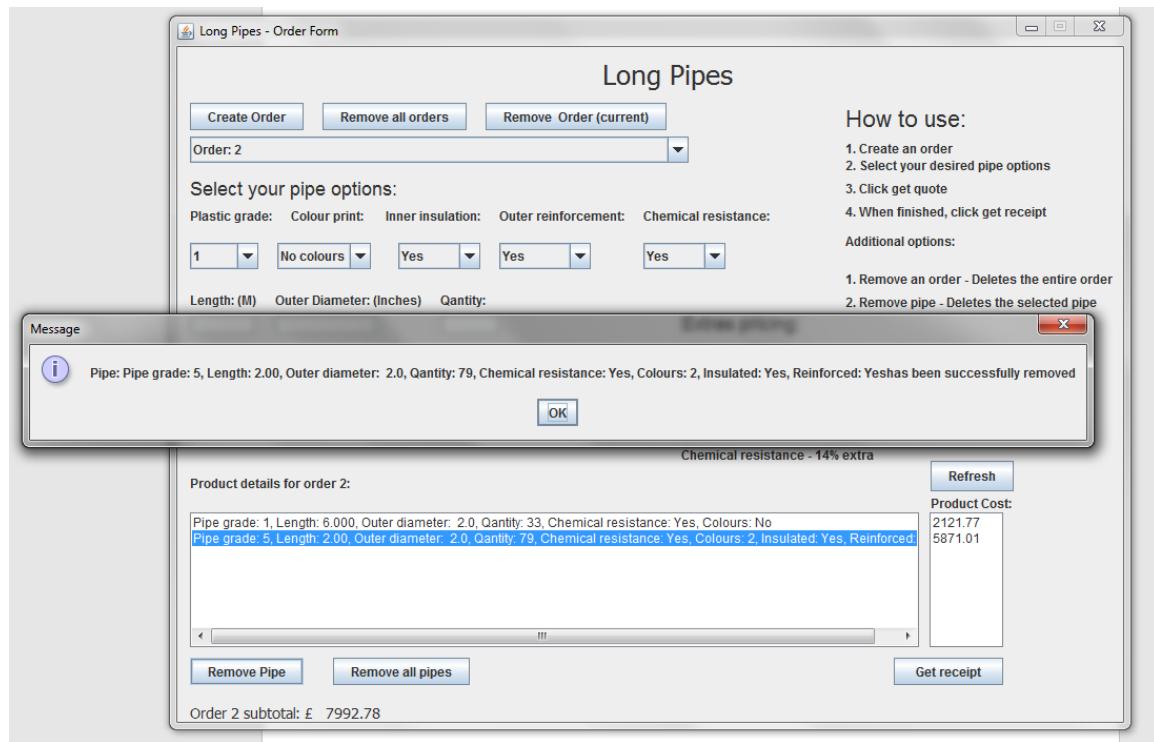
To make an amendment, please refer to the ordering screen.

Pipe grade: 3, Length: 1.00, Outer diameter: 1.0, Qantity: 1, Chemical resistance: Yes, Colours: 1  
 Pipe grade: 1, Length: 5.000, Outer diameter: 2.0, Qantity: 5, Chemical resistance: Yes, Colours: Nc  
 Pipe grade: 1, Length: 6.000, Outer diameter: 2.0, Qantity: 33, Chemical resistance: Yes, Colours: Nc  
 Pipe grade: 5, Length: 2.00, Outer diameter: 2.0, Qantity: 79, Chemical resistance: Yes, Colours: 2, Insulated: Yes  
 Pipe grade: 4, Length: 6.00, Outer diameter: 2.0, Qantity: 90, Chemical resistance: Yes, Colours: 2, Insulated: Yes

#Items ordered: 5 Total cost:  
 £ 25163.50

Thank you for shopping at Long Pipes

Next I will delete a product from order 2 and request a receipt.



Order 2 details:

## Long Pipes

Order: 2

Select your pipe options:

Plastic grade: 1 Colour print: No colours Inner insulation: Yes Outer reinforcement: Yes Chemical resistance: Yes

Length: (M) Outer Diameter: (Inches) Quantity:

Product details for order 2:

Pipe grade: 1, Length: 6.000, Outer diameter: 2.0, Quantity: 33, Chemical resistance: Yes, Colours: No

Order 2 cost: 2121.77

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

**Additional options:**

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

**Extras pricing:**

- 1 Colour - 12% extra
- 2 Colours - 16% extra
- Inner insulation - 13% extra
- Outer reinforcement - 17% extra
- Chemical resistance - 14% extra

**Product Cost:**  
2121.77

Now I will request a receipt.

## Long Pipes

Order: 2

Select your pipe options:

Plastic grade: 1 Colour print: No colours Inner insulation: Yes Outer reinforcement: Yes Chemical resistance: Yes

Length: (M) Outer Diameter: (Inches) Quantity:

Product details for order 2:

Pipe grade: 1, Length: 6.000, Outer diameter: 2.0, Quantity: 33, Chemical resistance: Yes, Colours: No

Order 2 cost: 2121.77

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

**Additional options:**

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

**Long Pipes**

Your order details:

This is the products that you have ordered from all of your order that you have made.

To make an amendment, please refer to the ordering screen.

Pipe grade: 3, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: Yes, Colours: 1  
 Pipe grade: 1, Length: 5.000, Outer diameter: 2.0, Quantity: 5, Chemical resistance: Yes, Colours: No  
 Pipe grade: 1, Length: 6.000, Outer diameter: 2.0, Quantity: 33, Chemical resistance: Yes, Colours: No  
 Pipe grade: 4, Length: 6.00, Outer diameter: 2.0, Quantity: 90, Chemical resistance: Yes, Colours: 2,

#Items ordered: 4 Total cost: £ 19292.49

Thank you for shopping at Long Pipes

Next I will delete all items from order 2.

### Long Pipes

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 2

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1	No colours	Yes	Yes	Yes
---	------------	-----	-----	-----

Length: (M) Outer Diameter: (Inches) Quantity:

--	--	--

[Get Quote](#)

**Extras pricing:**

Message

All pipes have been successfully removed.

[OK](#)

Product details for order 2:

Pipe grade: 1, Length: 6.000, Outer diameter: 2.0, Quantity: 33, Chemical resistance: Yes, Colours: No

[Refresh](#)

Product Cost: 2121.77

[Get receipt](#)

Order 2 cost: 2121.77

[Remove Pipe](#) [Remove all pipes](#)

After user clicks OK:

### Long Pipes

[Create Order](#) [Remove all orders](#) [Remove Order \(current\)](#)

Order: 2

Select your pipe options:

Plastic grade: Colour print: Inner insulation: Outer reinforcement: Chemical resistance:

1	No colours	Yes	Yes	Yes
---	------------	-----	-----	-----

Length: (M) Outer Diameter: (Inches) Quantity:

--	--	--

[Get Quote](#)

Product details for order 2:

No products have been ordered at the moment, please select the products details to continue

[Refresh](#)

Product Cost:

[Get receipt](#)

Order 2 cost: 0.00

[Remove Pipe](#) [Remove all pipes](#)

Next I will request a receipt.

**Long Pipes**

**Create Order**   **Remove all orders**   **Remove Order (current)**

Order: 3

**Select your pipe options:**

Plastic grade:  Colour print:  Inner insulation:  Outer reinforcement:  Chemical resistance:

Length: (M)  Outer Diameter: (Inches)  Quantity:

**Get Quote**

**Product details for order 3:**

Pipe grade: 4, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: Yes, Colours: No

#Items ordered: 3      Total cost: £ 17170.72

Thank you for shopping at Long Pipes

**Remove Pipe**   **Remove all pipes**   **Get receipt**

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote

- Deletes the entire order  
- Deletes the selected pipe

**Refresh**   **Product Cost:** 7.19

Order 3 subtotal: £ 16897.19

Now I will delete order 2 entirely:

**Long Pipes**

**Create Order**   **Remove all orders**   **Remove Order (current)**

Order: 2

**Select your pipe options:**

Plastic grade:  Colour print:  Inner insulation:  Outer reinforcement:  Chemical resistance:

Length: (M)  Outer Diameter: (Inches)  Quantity:

**Get Quote**

**Product details for order 2:**

No products have been ordered at the moment, please select the products details to continue

**Remove Pipe**   **Remove all pipes**   **Get receipt**

**How to use:**

1. Create an order
2. Select your desired pipe options
3. Click get quote
4. When finished, click get receipt

Additional options:

1. Remove an order - Deletes the entire order
2. Remove pipe - Deletes the selected pipe

**Extras pricing:**

10% extra  
13% extra  
17% extra  
14% extra

**Message**

Order: 2 has been successfully removed

**OK**

**Refresh**   **Product Cost:** No items

Order 2 subtotal: £ 0.00

Finally we will remove all orders.

**Long Pipes**

**Create Order**
**Remove all orders**
**Remove Order (current)**

Select your pipe options:  
 Plastic grade:  Colour print:  Inner insulation:  Outer reinforcement:  Chemical resistance:

Length: (M)
Outer Diameter: (Inches)
Quantity:

Extras pricing:  
 1 Colour - 12% extra  
 2 Colours - 16% extra  
 Inner insulation - 13% extra  
 Outer reinforcement - 17% extra  
 Chemical resistance - 14% extra

**Get Quote**

**Message**  
 All orders have been successfully removed

**OK**

Product details for order 1:  
 Pipe grade: 3, Length: 1.00, Outer diameter: 1.0, Quantity: 1, Chemical resistance: Yes, Colours: 1  
 Pipe grade: 1, Length: 5.000, Outer diameter: 2.0, Quantity: 5, Chemical resistance: Yes, Colours: No

**Refresh**
**Product Cost:**

5.63  
 267.90

**Remove Pipe**
**Remove all pipes**
**Get receipt**

Order 1 subtotal: £ 273.53

After user clicks OK:

**Long Pipes**

**Create Order**
**Remove all orders**
**Remove Order (current)**

Plastic grade:  Colour print:  Inner insulation:  Outer reinforcement:  Chemical resistance:

Length: (M)
Outer Diameter: (Inches)
Quantity:

1 Colour - 12% extra  
 2 Colours - 16% extra  
 Inner insulation - 13% extra  
 Outer reinforcement - 17% extra  
 Chemical resistance - 14% extra

**Get Quote**

**Unable to show product details for an order at the moment, consider making an order.**

**Refresh**

**Product Cost:**

5.63  
 267.90

**Remove Pipe**
**Remove all pipes**
**Get receipt**

**Unable to calculate order subtotal.**

Program is in original state.

## Appendix A.

### Source code

Main:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package coursework;
```

```
import javax.swing.JFrame;
import javax.swing.SwingUtilities;
```

```
/**
 *
 * @author up811216
 */
```

```
public class main {
```

```
    /**
     * @param c
     */
```

```
    public void runInteface(Interface c){
```

```
SwingUtilities.invokeLater(new Runnable() {  
    public void run() {  
        c.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        c.setResizable(false);  
        c.setVisible(true);  
    }  
});  
  
}  
  
/**  
 *  
 * @param c  
 */  
public void runReceipt(Receipt c){  
    SwingUtilities.invokeLater(new Runnable() {  
        public void run() {  
            c.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
            c.setResizable(false);  
            c.setVisible(false);  
        }  
    });  
}  
  
/**  
 *
```

```
* @param args
*/
public static void main(String[] args) {
    // TODO code application logic here

    main d = new main();
    OrderManage manag = new OrderManage(1);
    Receipt e = new Receipt(manag);
    Interface c = new Interface(d, manag, e);
    d.runInteface(c);
    d.runReceipt(e);

}

}

Interface:
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package coursework;
import javax.swing.DefaultListModel;
```

```
import java.util.*;  
  
import javax.swing.JOptionPane;  
  
/**  
 *  
 * @author up811216  
 */  
  
public class Interface extends javax.swing.JFrame {  
  
    /**  
     * Creates new form Interface  
     */  
  
    private main MainClass;  
  
    private OrderManage manager;  
  
    private Order order;  
  
    private int idCount=0;  
  
    private int Qantity=0;  
  
    private double Length=0;  
  
    private double OuterDiameter=0;  
  
    private Receipt receipt;  
  
    //private DefaultListModel li = new DefaultListModel();  
  
    /**  
     *  
     * @param mainClass  
     */
```

```
* @param Manager  
* @param rec  
*/  
  
public Interface(main mainClass, OrderManage Manager, Receipt rec) {  
  
    initComponents();  
  
    MainClass = mainClass;  
  
    manager = Manager;  
  
    receipt = rec;  
  
    orderListDropDownMenu.setVisible(false);  
  
    productList.add("No orders have been made at the moment, please make an order to  
continue and then select your product details");  
  
    orderSubtotalTitle.setText("Unable to calculate order subtotal at the moment.");  
  
    productDetailsTitle.setText("Unable to show product details for an order at the  
moment, consider making an order.");  
  
    orderContentsValueTitle.setVisible(false);  
  
    orderContentsTitle.setText("No orders have been made, please make an order.");  
  
    orderSubtotalValueTitle.setVisible(false);  
  
}  
  
/**  
 * This method is called from within the constructor to initialize the form.  
 * WARNING: Do NOT modify this code. The content of this method is always  
 * regenerated by the Form Editor.  
 */  
  
@SuppressWarnings("unchecked")  
// <editor-fold defaultstate="collapsed" desc="Generated Code">
```

```
private void initComponents() {  
  
    jDialog1 = new javax.swing.JDialog();  
  
    orderListDropDownMenu = new javax.swing.JComboBox<>();  
  
    createOrderBtn = new javax.swing.JButton();  
  
    removePipeBtn = new javax.swing.JButton();  
  
    chemicalResistanceTitle = new javax.swing.JLabel();  
  
    removeOrderBtn = new javax.swing.JButton();  
  
    getQuoteBtn = new javax.swing.JButton();  
  
    productList = new java.awt.List();  
  
    plasticGradeValueDropDownMenu = new javax.swing.JComboBox<>();  
  
    colorPrintValueDropDownMenu = new javax.swing.JComboBox<>();  
  
    innerInsulationValueDropDownMenu = new javax.swing.JComboBox<>();  
  
    outerReinforcementDropDownMenu = new javax.swing.JComboBox<>();  
  
    chemicalResistanceDropDownMenu = new javax.swing.JComboBox<>();  
  
    lengthTextField = new javax.swing.JTextField();  
  
    outerDiameterTextField = new javax.swing.JTextField();  
  
    quantityTextField = new javax.swing.JTextField();  
  
    plasticGradeTitle = new javax.swing.JLabel();  
  
    colorPrintTitle = new javax.swing.JLabel();  
  
    innerInsulatioTitle = new javax.swing.JLabel();  
  
    outerReinforcementTitle = new javax.swing.JLabel();  
  
    lengthTitle = new javax.swing.JLabel();  
  
    outerDiameterTitle = new javax.swing.JLabel();  
  
    quantityTitle = new javax.swing.JLabel();
```

```
companuTitle1 = new javax.swing.JLabel();

productValueList = new java.awt.List();

productDetailsTitle = new javax.swing.JLabel();

productCostTitle = new javax.swing.JLabel();

getReceiptBtn = new javax.swing.JButton();

orderSubtotalTitle = new javax.swing.JLabel();

orderSubtotalValueTitle = new javax.swing.JLabel();

extrasPricingTitle = new javax.swing.JLabel();

extrasPricingItem1 = new javax.swing.JLabel();

extrasPricingItem2 = new javax.swing.JLabel();

extrasPricingItem3 = new javax.swing.JLabel();

extrasPricingItem4 = new javax.swing.JLabel();

extrasPricingItem5 = new javax.swing.JLabel();

usageItem2Title = new javax.swing.JLabel();

usageItem1Title = new javax.swing.JLabel();

usageTitle = new javax.swing.JLabel();

usageItem3Title = new javax.swing.JLabel();

usageItem4Title = new javax.swing.JLabel();

additionalOptionsTitle = new javax.swing.JLabel();

additionalOptionsItem1 = new javax.swing.JLabel();

additionalOptionsItem2 = new javax.swing.JLabel();

selectPipeOptionsTitle = new javax.swing.JLabel();

removeAllOrdersBtn = new javax.swing.JButton();

removeAllPipesBtn = new javax.swing.JButton();

orderContentsTitle = new javax.swing.JLabel();
```

```
orderContentsValueTitle = new javax.swing.JLabel();

javax.swing.GroupLayout jDialog1Layout = new
javax.swing.GroupLayout(jDialog1.getContentPane());
jDialog1.getContentPane().setLayout(jDialog1Layout);
jDialog1Layout.setHorizontalGroup(
jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGap(0, 400, Short.MAX_VALUE)
);
jDialog1Layout.setVerticalGroup(
jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGap(0, 265, Short.MAX_VALUE)
);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Long Pipes - Order Form");

orderListDropDownMenu.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        orderListDropDownMenuActionPerformed(evt);
    }
});

createOrderBtn.setText("Create Order");
```

```
createOrderBtn.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        createOrderBtnActionPerformed(evt);  
    }  
});  
  
removePipeBtn.setText("Remove Pipe");  
removePipeBtn.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        removePipeBtnActionPerformed(evt);  
    }  
});  
  
chemicalResistanceTitle.setText("Chemical resistance:");  
  
removeOrderBtn.setText("Remove Order (current)");  
removeOrderBtn.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        removeOrderBtnActionPerformed(evt);  
    }  
});  
  
getQuoteBtn.setText("Get Quote");  
getQuoteBtn.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        getQuoteBtnActionPerformed(evt);
    }

});

plasticGradeValueDropDownMenu.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "1", "2", "3", "4", "5" }));

colorPrintValueDropDownMenu.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "No colours", "1 colour", "2 colours"
}));

innerInsulationValueDropDownMenu.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "Yes", "No" }));

outerReinforcementDropDownMenu.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "Yes", "No" }));

chemicalResistanceDropDownMenu.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "Yes", "No" }));

lengthTextField.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        lengthTextFieldActionPerformed(evt);
    }
});

plasticGradeTitle.setText("Plastic grade:");
```

```
colorPrintTitle.setText("Colour print:");

innerInsulatioTitle.setText("Inner insulation:");

outerReinforcementTitle.setText("Outer reinforcement:");

lengthTitle.setText("Length: (M)");

outerDiameterTitle.setText("Outer Diameter: (Inches)");

qantityTitle.setText("Qantity:");

companuTitle1.setFont(new java.awt.Font("Tahoma", 0, 26)); // NOI18N
companuTitle1.setText("Long Pipes");

productDetailsTitle.setText("Product details:");

productCostTitle.setText("Product Cost: (£)");

getReceiptBtn.setText("Get receipt");
getReceiptBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        getReceiptBtnActionPerformed(evt);
    }
});
```

```
orderSubtotalTitle.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
orderSubtotalTitle.setText("Order Cost: £");
```

```
orderSubtotalValueTitle.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
orderSubtotalValueTitle.setText("o.oo");
```

```
extrasPricingTitle.setFont(new java.awt.Font("Lucida Grande", 0, 18)); // NOI18N  
extrasPricingTitle.setText("Extras pricing: ");
```

```
extrasPricingItem1.setText("1 Colour - 12% extra");
```

```
extrasPricingItem2.setText("2 Colours - 16% extra ");
```

```
extrasPricingItem3.setText("Inner insulation - 13% extra");
```

```
extrasPricingItem4.setText("Outer reinforcement - 17% extra ");
```

```
extrasPricingItem5.setText("Chemical resistance - 14% extra");
```

```
usageItem2Title.setText("2. Select your desired pipe options");
```

```
usageItem1Title.setText("1. Create an order");
```

```
usageTitle.setFont(new java.awt.Font("Lucida Grande", 0, 22)); // NOI18N
```

```
usageTitle.setText("How to use:");

usageItem3Title.setText("3. Click get quote");

usageItem4Title.setText("4. When finished, click get receipt");

additionalOptionsTitle.setText("Additional options:");

additionalOptionsItem1.setText("1. Remove an order - Deletes the entire order");

additionalOptionsItem2.setText("2. Remove pipe - Deletes the selected pipe");

selectPipeOptionsTitle.setFont(new java.awt.Font("Lucida Grande", 0, 18)); // NOI18N
selectPipeOptionsTitle.setText("Select your pipe options:");

removeAllOrdersBtn.setText("Remove all orders");
removeAllOrdersBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        removeAllOrdersBtnActionPerformed(evt);
    }
});

removeAllPipesBtn.setText("Remove all pipes");
removeAllPipesBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```



```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(orderListDropDownMenu,
javax.swing.GroupLayout.PREFERRED_SIZE, 474,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addGroup(layout.createSequentialGroup()
              .addComponent(createOrderBtn)
              .addGap(18, 18, 18)
              .addComponent(removeAllOrdersBtn)
              .addGap(18, 18, 18)
              .addComponent(removeOrderBtn)))
          .addGap(0, 0, Short.MAX_VALUE))
          .addGroup(layout.createSequentialGroup())
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(layout.createSequentialGroup()
                  .addComponent(orderSubtotalTitle)
                  .addComponent(orderSubtotalValueTitle))
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
                  .addGroup(layout.createSequentialGroup()
                      .addComponent(createOrderBtn)
                      .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
                      .addComponent(removeAllOrdersBtn)
                      .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
                      .addComponent(removeOrderBtn)))
                  .addGroup(layout.createSequentialGroup()
                      .addComponent(orderSubtotalTitle)
                      .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
                      .addComponent(orderSubtotalValueTitle))))))))
```

```
.addComponent(productList,
javax.swing.GroupLayout.PREFERRED_SIZE, 693,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))

.addComponent(layout.createSequentialGroup()

.addComponent(productDetailsTitle)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(productCostTitle)

.addComponent(productValueList,
javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addComponent(removePipeBtn)

.addGap(28, 28, 28)

.addComponent(removeAllPipesBtn)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(getReceiptBtn,
javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addComponent(layout.createSequentialGroup()

.addComponent(selectPipeOptionsTitle)
```

```
.addComponent(getQuoteBtn)  
.addGroup(layout.createSequentialGroup()  
    .addComponent(orderContentsTitle)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
    .addComponent(orderContentsValueTitle))  
    .addGroup(layout.createSequentialGroup())  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(layout.createSequentialGroup())  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,  
false)  
    .addComponent(lengthTextField)  
    .addComponent(lengthTitle))  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(layout.createSequentialGroup()  
        .addGap(18, 18, 18)  
        .addComponent(outerDiameterTitle))  
    .addGroup(layout.createSequentialGroup()  
        .addGap(18, 18, 18)  
        .addComponent(outerDiameterTextField,  
javax.swing.GroupLayout.PREFERRED_SIZE, 97,  
javax.swing.GroupLayout.PREFERRED_SIZE))))  
    .addGroup(layout.createSequentialGroup())  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(layout.createSequentialGroup()
    .addComponent(plasticGradeTitle)
    .addGap(18, 18, 18)
    .addComponent(colorPrintTitle))
    .addGroup(layout.createSequentialGroup()
        .addComponent(plasticGradeValueDropDownMenu,
        javax.swing.GroupLayout.PREFERRED_SIZE, 65,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(colorPrintValueDropDownMenu,
        javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(innerInsulatioTitle)
    .addComponent(innerInsulationValueDropDownMenu,
    javax.swing.GroupLayout.PREFERRED_SIZE, 78,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(qantityTextField,
        javax.swing.GroupLayout.PREFERRED_SIZE, 56,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(qantityTitle)
        .addGroup(layout.createSequentialGroup()
            .addGap(27, 27, 27)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(outerReinforcementTitle)
```

```
.addComponent(outerReinforcementDropDownMenu,  
javax.swing.GroupLayout.PREFERRED_SIZE, 87,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
  
.addGap(18, 18, 18)  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
    .addComponent(chemicalResistanceDropDownMenu,  
javax.swing.GroupLayout.PREFERRED_SIZE, 78,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
    .addComponent(chemicalResistanceTitle))))))  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 70,  
Short.MAX_VALUE)  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
    .addComponent(additionalOptionsTitle)  
  
    .addComponent(additionalOptionsItem1)  
  
    .addComponent(additionalOptionsItem2)  
  
    .addComponent(usageItem1Title)  
  
    .addComponent(usageItem2Title,  
javax.swing.GroupLayout.PREFERRED_SIZE, 250,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
    .addComponent(usageItem3Title)  
  
    .addComponent(usageItem4Title)  
  
    .addComponent(usageTitle)  
  
    .addComponent(extrasPricingItem3)  
  
    .addComponent(extrasPricingItem5)  
  
    .addComponent(extrasPricingItem2)  
  
    .addComponent(extrasPricingItem1)
```

```
.addComponent(extrasPricingItem4)
.addComponent(extrasPricingTitle)))))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))))

);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(0, 4, Short.MAX_VALUE)
        .addComponent(companuTitle1)
        .addGap(11, 11, 11)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(usageTitle)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(usageItem1Title)
    .addGap(0, 0, 0)
    .addComponent(usageItem2Title)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(usageItem3Title)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(usageItem4Title)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
.addComponent(additionalOptionsTitle)  
.addGap(20, 20, 20)  
.addComponent(additionalOptionsItem)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
.addComponent(additionalOptionsItem2)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
.addComponent(extrasPricingTitle)  
.addGap(7, 7, 7)  
.addComponent(extrasPricingItem1)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
.addComponent(extrasPricingItem2)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
.addComponent(extrasPricingItem3)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
.addComponent(extrasPricingItem4)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
.addComponent(extrasPricingItem5)  
.addGap(18, 21, Short.MAX_VALUE))  
.addGroup(layout.createSequentialGroup())  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE))  
.addComponent(createOrderBtn)
```

```
.addComponent(removeAllOrdersBtn)  
addComponent(removeOrderBtn)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
addComponent(orderListDropDownMenu,  
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addGap(12, 12, 12)  
addComponent(selectPipeOptionsTitle,  
javax.swing.GroupLayout.PREFERRED_SIZE, 24,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
addGroup(layout.createSequentialGroup())  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
addGroup(layout.createSequentialGroup())  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  
addComponent(plasticGradeTitle)  
addComponent(colorPrintTitle))  
addGap(18, 18, 18)  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  
addComponent(plasticGradeValueDropDownMenu,  
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addComponent(colorPrintValueDropDownMenu,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

.addComponent(innerInsulatioTitle)

.addGap(18, 18, 18)

.addComponent(innerInsulationValueDropDownMenu,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(outerReinforcementTitle)

.addGap(18, 18, 18)

.addComponent(outerReinforcementDropDownMenu,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(21, 21, 21)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lengthTitle)

.addComponent(outerDiameterTitle)

.addComponent(qantityTitle))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lengthTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(outerDiameterTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(qantityTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(27, 27, 27)

        .addComponent(getQuoteBtn)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(orderContentsTitle)
        .addComponent(orderContentsValueTitle))
        .addGap(26, 26, 26))

.addGroup(layout.createSequentialGroup()
        .addComponent(chemicalResistanceTitle)
        .addGap(18, 18, 18)
        .addComponent(chemicalResistanceDropDownMenu,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(productCostTitle)
        .addComponent(productDetailsTitle))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 13,
Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(productList, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(productValueList,
javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(10, 10, 10)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(removePipeBtn)
        .addComponent(removeAllPipesBtn))
        .addGap(25, 25, 25)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(orderSubtotalTitle)
        .addComponent(orderSubtotalValueTitle)))

    .addGroup(layout.createSequentialGroup()
        .addGap(19, 19, 19)
        .addComponent(getReceiptBtn))

    .addContainerGap())
);

});
```

```
    pack();  
    } // </editor-fold>
```

```
/*
```

This method will check range, type and combinational errors that can occur with the Text fields

```
*/
```

```
/**  
 *  
 * @param length  
 * @param outerDiameter  
 * @param quantity  
 * @return  
 */
```

```
public boolean VerifyInputs(String length, String outerDiameter, String quantity){  
    boolean valid=false, lengthValid=false, outerDiameterValid=false, quantityValid=false,  
    lengthParsed=false, outerDiameterParsed=false;
```

```
    //If each text field has a value then proceed
```

```
    if(length.length()>0 && outerDiameter.length()>0 && quantity.length()>0){
```

```
        //Trying each text field to see if it can be converted to the required type
```

```
        try{
```

```
            Length = Double.parseDouble(length);
```

```
            lengthParsed=true;
```

```
lengthValid=true;

}

catch(NumberFormatException ex){

    //Providing the appropriate error message for the type error

    JOptionPane.showMessageDialog(null, "Please enter a number for the length,
letters and characters are not allowed");

    lengthValid=false;

}

try{

    Qantity = Integer.parseInt(qantity);

    outerDiameterValid=true;

}

catch(NumberFormatException ex){

    JOptionPane.showMessageDialog(null, "Please enter a number for the qantity,
letters and characters are not allowed");

    outerDiameterValid=false;

}

try{

    OuterDiameter = Double.parseDouble(outerDiameter);

    outerDiameterParsed=true;

}
```

```
    qantityValid=true;

}

catch(NumberFormatException ex){

    JOptionPane.showMessageDialog(null, "Please enter a number for the outer
diameter, letters and characters are not allowed");

    qantityValid=false;

}

//Checking that the string does not contain reserved letters

if (lengthValid && lengthParsed==true){

    if((length.contains("f")==true || length.contains("d")==true)){

        JOptionPane.showMessageDialog(null, "Please enter a number for the length,
letters and characters are not allowed");

        lengthValid=false;

    }

}

if(outerDiameterValid && outerDiameterParsed==true){

    if((outerDiameter.contains("f")==true || outerDiameter.contains("d")==true)){

        JOptionPane.showMessageDialog(null, "Please enter a number for the outer
diameter, letters and characters are not allowed");

        outerDiameterValid=false;

    }

}
```

```
if(quantityValid && lengthValid && outerDiameterValid){  
    valid=true;  
}  
  
//If the type errors have passed then proceed to range checks  
  
if(valid){  
    //Checking the rangers of the values  
  
    if(!(Length>=0.25 && Length<=6)){  
        valid = false;  
  
        JOptionPane.showMessageDialog(null, "Please enter a valid length, the  
length must greater than 0.25M and no greater than 6M");  
    }  
  
    if(!(Qantity>0 && Qantity<=100)){  
        valid = false;  
  
        JOptionPane.showMessageDialog(null, "Please enter a valid qantity, the  
qantity must greater than 0 and no greater than 100");  
    }  
  
    if(!(OuterDiameter>=0.25&& OuterDiameter<=5)){  
        valid = false;  
  
        JOptionPane.showMessageDialog(null, "Please enter a valid outer diameter,  
the outer diameter must 0.25 or Inches greater and no greater than 5 Inches");  
    }  
  
    return valid;  
}
```

```
}

//Checking combinations of possible errors

else if(length.length()>o && outerDiameter.length()>o){

    JOptionPane.showMessageDialog(null, "Please enter a quantity, no qantity
found");

}

else if(qantity.length()>o && outerDiameter.length()>o){

    JOptionPane.showMessageDialog(null, "Please enter a length, no length found");

}

else if(qantity.length()>o && length.length()>o){

    JOptionPane.showMessageDialog(null, "Please enter a outer diameter, no outer
diameter found");

}

else if(qantity.length()==o && outerDiameter.length()==o && length.length()>o){

    JOptionPane.showMessageDialog(null, "Please enter a qantity and outer diameter,
no quantity or outer diameter found");

}

else if(qantity.length()==o && length.length()==o && outerDiameter.length()>o){

    JOptionPane.showMessageDialog(null, "Please enter a length and qantity, no
length and qantity found");

}

else if(length.length()==o && outerDiameter.length()==o && qantity.length()>o){

    JOptionPane.showMessageDialog(null, "Please enter a length and outer diameter,
no length and outer diamater found");

}

else {

    JOptionPane.showMessageDialog(null, "Please enter a length, outer diameter and
qantity to proceed");
}
```

```
    }

    //If the range, type and combinational errors have passed then true will be returned
    overall
```

```
    return valid;
```

```
}
```

```
/*
```

This is the create order button, it will first check that if the orderListDropDownMneu is not visible, then it will make it visible.

Orders are added to the manager's list of order.

```
*/
```

```
private void createOrderBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    if(orderListDropDownMenu.isVisible()==false){
```

```
        orderListDropDownMenu.setVisible(true);
```

```
        Order ord = new Order(idCount);
```

```
        manager.createOrder(ord);
```

```
        //Adding the order to the orderListDropDownMenu and assigning an ID
```

```
        orderListDropDownMenu.addItem(String.format("Order: %d",
manager.getOrder(idCount).getID()+1));
```

```
        idCount+=1;
```

```
        JOptionPane.showMessageDialog(null, "Successfully created a new order, please
proceed");
```

```
        //If there are no items, then remove all products in the productList and reset this
list
```

```
if(manager.getOrder(orderListDropDownMenu.getSelectedIndex()).getPipeItemListSize()
<=0){

    productList.removeAll();

    productList.add("No products have been ordered at the moment, please select the
products details to continue");

}

//Changing the subtotal to the first order value

orderContentsValueTitle.setText(""+(orderListDropDownMenu.getSelectedIndex()+1));

}

else{

    //Adding an item when the orderList appears

    Order ord = new Order(idCount);

    manager.createOrder(ord);

    orderListDropDownMenu.addItem(String.format("Order: %d",
manager.getOrder(idCount).getID()+1));

    idCount+=1;

    JOptionPane.showMessageDialog(null, "Successfully created a new order, please
proceed");

}

//Display no products in the newly created order

if(manager.getOrder(orderListDropDownMenu.getSelectedIndex()).getPipeItemListSize()
==0){

    if(productList.getItemCount()==0){

        productList.add("No products to display at the moment");
}
```

```
productValueList.add("No items");

}

}

if(!orderSubtotalValueTitle.isVisible()){

    orderSubtotalValueTitle.setVisible(true);

}

//Setting the lables to this orders values

productDetailsTitle.setText(String.format("Product details for order %d:",
orderListDropDownMenu.getSelectedIndex()+1));

    orderSubtotalTitle.setText(String.format("Order %d subtotal: £",
orderListDropDownMenu.getSelectedIndex()+1));

orderSubtotalValueTitle.setText(manager.getOrder(orderListDropDownMenu.getSelected
Index()).getOrderCost());

if(!orderContentsValueTitle.isVisible())

    orderContentsValueTitle.setVisible(true);

orderContentsTitle.setText("Viewing order contents for order: ");

}

//This method is used when the user clicks on the ordewrList to view a different order

private void orderListDropDownMenuActionPerformed(java.awt.event.ActionEvent
evt) {

    // TODO add your handling code here:

    //If the user has clicked then proceed

    if(orderListDropDownMenu.hasFocus()){

        Order ord = manager.getOrder(orderListDropDownMenu.getSelectedIndex());
}
```

```
productList.removeAll();

productValueList.removeAll();

//Resetting all labels to the new order

productDetailsTitle.setText(String.format("Product details for order %d:",
orderListDropDownMenu.getSelectedIndex()+1));

orderSubtotalTitle.setText(String.format("Order %d subtotal: £",
orderListDropDownMenu.getSelectedIndex()+1));

orderSubtotalValueTitle.setText(ord.getOrderCost());

orderContentsTitle.setVisible(true);

orderContentsValueTitle.setVisible(true);

orderContentsValueTitle.setText(""+(orderListDropDownMenu.getSelectedIndex()+1));

//reading all procts from the newly selected order into the productList

if(ord.getPipeItemListSize()>0){

for(Pipe i: ord.getPipeList()){

productList.add(i.getPipeInfoSimplified());

productValueList.add(i.getPipeCostStr());

}

}

else{

//If there are no products, we must display this to the user

productList.add("No products have been ordered at the moment, please select the
products details to continue");

productValueList.add("No items");

}

}
```

}

```
private void lengthTextFieldActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

}

/\*

This method is called when the user clicks on the remove order button and will remove  
the currently selected order

\*/

```
private void removeOrderBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    //Check that the size of the currently selected order has products to be removed  
    if(manager.getOrderListSize()>0){  
        JOptionPane.showMessageDialog(null,  
        (orderListDropDownMenu.getItemAt(orderListDropDownMenu.getSelectedIndex()) + "  
        has been successfully removed");  
        manager.RemoveOrder(orderListDropDownMenu.getSelectedIndex());  
  
        //jComboBox1.removeItem(jComboBox1.getItemAt(jComboBox1.getSelectedIndex()));  
  
        orderListDropDownMenu.removeItemAt(orderListDropDownMenu.getSelectedIndex());  
        idCount =0;  
        int tempSize = orderListDropDownMenu.getItemCount();
```

```
if (tempSize>o){

    orderListDropDownMenu.removeAllItems();

    //Changin the ID's of the orders as these need to be reconfigured when the are read
    back by orderList

    for(int i =0; i<tempSize; i++){

        manager.getOrder(i).changeID(i);

        orderListDropDownMenu.addItem(String.format("Order: %d",
        manager.getOrder(idCount).getID()+1));

        idCount+=1;

    }

    //The productList is updated

    refresh();

    //Resetting labels

    productDetailsTitle.setText(String.format("Product details for order %d:",
    orderListDropDownMenu.getSelectedIndex()+1));

    orderSubtotalTitle.setText(String.format("Order %d subtotal: £",
    orderListDropDownMenu.getSelectedIndex()+1));

    orderSubtotalValueTitle.setText(manager.getOrder(orderListDropDownMenu.getSelected
    Index()).getOrderCost());

    orderContentsValueTitle.setText(""+
    (orderListDropDownMenu.getSelectedIndex()+1));

}

//The recceipt is updated

checkReceipt();
```

//If ther are no orders that have been made, we need to reset the labels to their orginal- inital run-time state

```
if(orderListDropDownMenu.getItemCount()==0){  
    receipt.setVisible(false);  
    productDetailsTitle.setText("Unable to show product details for an order at the moment, consider making an order.");  
    orderSubtotalTitle.setText("Unable to calculate order subtotal.");  
    orderSubtotalValueTitle.setText("0.00");  
    orderSubtotalValueTitle.setVisible(false);  
    orderListDropDownMenu.setVisible(false);  
    orderContentsTitle.setText("No orders have been made, please make an order.");  
    orderContentsValueTitle.setVisible(false);  
    productList.add("No orders have been made at the moment, please make an order to continue and then select your product details");  
    productList.removeAll();  
    productList.removeAll();  
    idCount=0;  
}  
}  
else{  
    JOptionPane.showMessageDialog(null, "Unable to remove a selected order, no orders have been made, please make an order to proceed");  
}  
}  
}  
//This method will check if the receipt can be shown and if it is alrwady shown, we need to update its productsList
```

```
/**  
 *  
 */  
  
public void checkReceipt(){  
  
    if(receipt.isVisible()){  
        //Populate Components will update our productsList  
        receipt.populateComponents();  
        //If there are no items from any orders  
        if(receipt.getNumCount()==0){  
            JOptionPane.showMessageDialog(null, "Receipt window will now close as there  
are no items that have been ordered in any order, please add a product to continue.");  
            receipt.setVisible(false);  
        }  
        else{  
            //Providn the update to the user  
            JOptionPane.showMessageDialog(null, "Your receipt has now been updated.");  
        }  
    }  
}  
  
/*  
 This is the remove pipe method. It will remove the ucrrently selected pipe form the  
product list, update the receipt if it is shown  
*/  
  
private void removePipeBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

```
// TODO add your handling code here:  
  
//If there are orders, then proceed  
  
if(manager.getOrderListSize()>0){  
  
    //If the user has selected a product then proceed  
  
    if(productList.getSelectedIndex()>=0){  
  
        //If there are products in the productList then proceed  
  
        if(productList.getItemCount()>=0){  
  
            if(manager.getOrder(orderListDropDownMenu.getSelectedIndex()).getPipeList().size()>0)  
            {  
  
                JOptionPane.showMessageDialog(null, "##--" +  
manager.getOrder(orderListDropDownMenu.getSelectedIndex()).getPipeList().get(produ  
ctList.getSelectedIndex()).getPipeInfoSimplified() + "--##" + " has been successfully  
removed from Order: " + (orderListDropDownMenu.getSelectedIndex()+1));  
  
                //Removing the pipe from the correct order  
  
                manager.getOrder(orderListDropDownMenu.getSelectedIndex()).getPipeList().remove(pr  
oductList.getSelectedIndex());  
  
                //If the order has no products to show  
  
                if(productList.getItemCount()>0){  
  
                    productList.remove(productList.getSelectedIndex());  
  
                    productList.remove(productList.getSelectedIndex());  
  
                }  
  
                //Update the productsList  
  
                refresh();  
  
                //Update receipt
```

```
checkReceipt();

orderSubtotalValueTitle.setText(manager.getOrder(orderListDropDownMenu.getSelectedIndex()).getOrderCost());
}

else{

    JOptionPane.showMessageDialog(null, "No products to display at the moment,
please add an item to remove");

}

}

else{

    JOptionPane.showMessageDialog(null, "No products to display at the moment,
please add an item to remove");

}

}

else{

    JOptionPane.showMessageDialog(null, "No pipe selected - please select a pipe to
continue.");

}

}

else{

    JOptionPane.showMessageDialog(null, "Unable to remove selected pipe - No
orders have been made, please make an order an order and select a pipe to remove to
proceed");

}

}

}
```

```
/*
```

This is the getQuote button which will call verifyInputs to check if the inputs are accepted and then attempt to add a pipe

```
*/
```

```
private void getQuoteBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    boolean ordered=false;  
  
    //If there are orders then proceed  
  
    if(manager.getOrderListSize()>0){  
  
        //If verifyInputs returns true then proceed  
  
        if(VerifyInputs(lengthTextField.getText(), outerDiameterTextField.getText(),  
        quantityTextField.getText())){  
  
            //Converting chemicalResistance, outerReinforcement and outerDiameter drop  
            //down menus to their boolean equivalents  
  
            boolean resist;  
  
            if(chemicalResistanceDropDownMenu.getSelectedIndex()==0){  
  
                resist=true;  
  
            }  
  
            else{  
  
                resist=false;  
  
            }  
  
            boolean reinforced;  
  
            if(outerReinforcementDropDownMenu.getSelectedIndex()==0){  
  
                reinforced=true;  
  
            }
```

```
    }

    else{
        reinforced=false;

    }

    boolean insulation;

    if(innerInsulationValueDropDownMenu.getSelectedIndex()==0){

        insulation=true;

    }

    else{

        insulation=false;

    }

}
```

//Searching for the required type of pipe based on the users' defined properties

```
if((plasticGradeValueDropDownMenu.getSelectedIndex()==0 ||
plasticGradeValueDropDownMenu.getSelectedIndex()==1 ||
plasticGradeValueDropDownMenu.getSelectedIndex()==2)&&
(colorPrintValueDropDownMenu.getSelectedIndex()==0 && (reinforced==false &&
insulation==false))){
```

```
    Type1 t1 = new Type1(Length, OuterDiameter, Qantity, resist,
plasticGradeValueDropDownMenu.getSelectedIndex()+1,
manager.getOrder(orderListDropDownMenu.getSelectedIndex()));

    manager.getOrder(orderListDropDownMenu.getSelectedIndex()).addPipe(t1);

    JOptionPane.showMessageDialog(null, "Pipe has been sucessfully added, please
continue.");

    ordered=true;

}

else if ((plasticGradeValueDropDownMenu.getSelectedIndex()==1 ||
plasticGradeValueDropDownMenu.getSelectedIndex()==2 ||
plasticGradeValueDropDownMenu.getSelectedIndex()==3) &&
(colorPrintValueDropDownMenu.getSelectedIndex()==1 && (reinforced==false &&
insulation==false))){

    Type2 t2 = new Type2(Length, OuterDiameter, Qantity, resist,
plasticGradeValueDropDownMenu.getSelectedIndex()+1,
manager.getOrder(orderListDropDownMenu.getSelectedIndex()));

    manager.getOrder(orderListDropDownMenu.getSelectedIndex()).addPipe(t2);

    JOptionPane.showMessageDialog(null, "Pipe has been sucessfully added, please
continue.");

    ordered=true;

}

else if ((plasticGradeValueDropDownMenu.getSelectedIndex()==1 ||
plasticGradeValueDropDownMenu.getSelectedIndex()==2 ||
plasticGradeValueDropDownMenu.getSelectedIndex()==3 ||
plasticGradeValueDropDownMenu.getSelectedIndex()==4) &&
(colorPrintValueDropDownMenu.getSelectedIndex()==2 && (reinforced==false &&
insulation==false)) ){

    Type3 t3 = new Type3(Length, OuterDiameter, Qantity, resist,
plasticGradeValueDropDownMenu.getSelectedIndex()+1,
manager.getOrder(orderListDropDownMenu.getSelectedIndex()));

    manager.getOrder(orderListDropDownMenu.getSelectedIndex()).addPipe(t3);
```

```
JOptionPane.showMessageDialog(null, "Pipe has been sucessfully added, please  
continue.");  
  
        ordered=true;  
  
    }  
  
    else if ((plasticGradeValueDropDownMenu.getSelectedIndex()==1 ||  
plasticGradeValueDropDownMenu.getSelectedIndex()==2 ||  
plasticGradeValueDropDownMenu.getSelectedIndex()==3 ||  
plasticGradeValueDropDownMenu.getSelectedIndex()==4) &&  
(colorPrintValueDropDownMenu.getSelectedIndex()==2) && (reinforced==false &&  
insulation==true)) {  
  
    Type4 t4 = new Type4(Length, OuterDiameter, Qantity, resist,  
plasticGradeValueDropDownMenu.getSelectedIndex()+1,  
manager.getOrder(orderListDropDownMenu.getSelectedIndex()));  
  
    manager.getOrder(orderListDropDownMenu.getSelectedIndex()).addPipe(t4);  
  
    JOptionPane.showMessageDialog(null, "Pipe has been sucessfully added, please  
continue.");  
  
    ordered=true;  
  
}  
  
else if ((plasticGradeValueDropDownMenu.getSelectedIndex()==2 ||  
plasticGradeValueDropDownMenu.getSelectedIndex()==3 ||  
plasticGradeValueDropDownMenu.getSelectedIndex()==4) &&  
(colorPrintValueDropDownMenu.getSelectedIndex()==2) &&  
colorPrintValueDropDownMenu.getSelectedIndex()==2 && (reinforced==true &&  
insulation==true))) {  
  
    Type5 t5 = new Type5(Length, OuterDiameter, Qantity, resist,  
plasticGradeValueDropDownMenu.getSelectedIndex()+1,  
manager.getOrder(orderListDropDownMenu.getSelectedIndex()));  
  
    manager.getOrder(orderListDropDownMenu.getSelectedIndex()).addPipe(t5);  
  
    JOptionPane.showMessageDialog(null, "Pipe has been sucessfully added, please  
continue.");  
  
    ordered=true;
```

```
    }

    else{

        //If no type of pipe can be found then we must tell the user

        JOptionPane.showMessageDialog(null, "We do not supply this type of pipe
based on your properties, please try again.");

    }

}

}

else{

    JOptionPane.showMessageDialog(null, "No orders have been created yet, please
create an order");

}

//If an order has been made, we need to reset the dropdown menus and text fields
and change the orderCost for the subtotal

if(ordered){

    refresh();

    plasticGradeValueDropDownMenu.setSelectedIndex(o);

    colorPrintValueDropDownMenu.setSelectedIndex(o);

    innerInsulationValueDropDownMenu.setSelectedIndex(o);

    outerReinforcementDropDownMenu.setSelectedIndex(o);

    chemicalResistanceDropDownMenu.setSelectedIndex(o);

    lengthTextField.setText("");

    outerDiameterTextField.setText("");

}
```

```
qantityTextField.setText("");
ordered=false;

orderSubtotalValueTitle.setText(manager.getOrder(orderListDropDownMenu.getSelected
Index()).getOrderCost());

//Checking the receipt

if(receipt.isVisible()){

receipt.populateComponents();

JOptionPane.showMessageDialog(null, "Your receipt has now been updated");

}

}

}

/*  
 Refresh method will update the products List when there is a change i.e. adding a new  
 pipe or removing a pipe  
 */  
  
/**  
 *  
 */
```

```
public void refresh(){

if(orderListDropDownMenu.isVisible()){

Order ord = manager.getOrder(orderListDropDownMenu.getSelectedIndex());
```

```
productList.removeAll();

productValueList.removeAll();

//Reading the products again into productList

if(ord.getPipeItemListSize()>0){

for(Pipe i: ord.getPipeList()){

    productList.add(i.getPipeInfoSimplified());

    productValueList.add(i.getPipeCostStr());

}

}

else{

    //JOptionPane.showMessageDialog(null, "No items to display, please add a pipe");

    productList.add("No products have been ordered at the moment, please select the products details to continue");

}

//Resetting the orderSubtotal values and titles

orderSubtotalTitle.setText(String.format("Order %d subtotal: £",
orderListDropDownMenu.getSelectedIndex()+1));

orderSubtotalValueTitle.setText(manager.getOrder(orderListDropDownMenu.getSelectedIndex()).getOrderCost());

}

/*
Get Receipt will check that a receipt can be displayed, if it is already display, we msut tell the user

*/
```

```
private void getReceiptBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    boolean minProductLimit = false;  
  
    if(manager.getOrderListSize()>=1){  
  
        if(!receipt.isVisible()){  
  
            for(int i = 0; i<manager.getOrderListSize(); i++){  
  
                //If the minProductLimit is false at the first instance of a product being found for  
                any order then we can set this to true and hshow the Receipt  
  
                if(manager.getOrder(i).getPipeItemListSize()>=1 && minProductLimit==false){  
  
                    receipt = new Receipt(manager);  
  
                    receipt.setResizable(false);  
  
                    receipt.setVisible(true);  
  
                    minProductLimit = true;  
  
                }  
  
                //If no products for any order has not been found, we display the appropiate  
                message  
  
                else if(minProductLimit==false &&  
manager.getOrder().indexOf(manager.getOrder(i))==manager.getOrder().size()-1){  
  
                    JOptionPane.showMessageDialog(null, "No products have been found for any  
order, please order a pipe to continue.");  
  
                }  
  
                //Reset minProductLimit for next time use  
  
                minProductLimit = false;  
  
            }else{  
  
                JOptionPane.showMessageDialog(null, "The receipt is already displayed, please  
view/update your receipt.");  
  
            }  
    }  
}
```

```
    }

}

else{
    //No orders have been made yet

    JOptionPane.showMessageDialog(null, "Unable to display receipt, no orders have
been made at the moment");

}

/*  
Removing all orders from the order List and resetting the labels as required  
*/  
  
private void removeAllOrdersBtnActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    //If there are orders then proceed i.e. we can now remove all orders  
  
    if(manager.getOrderListSize()>0){  
  
        //Looping to remove all orders  
  
        for(int i=0; i<orderListDropDownMenu.getItemCount(); i++){  
  
            manager.RemoveOrder(o);  
  
            //System.out.println(manager.getOrderListSize());  
        }  
  
        //jComboBox1.removeItem(jComboBox1.getItemAt(jComboBox1.getSelectedIndex()));  
  
        //Resetting the dropDownMenus and text fields  
  
        orderListDropDownMenu.removeAllItems();
```

```
lengthTextField.setText("");
outerDiameterTextField.setText("");
qantityTextField.setText("");
if(receipt.isVisible()){

    JOptionPane.showMessageDialog(null, "Receipt window will now close as there
are no orders, please make an order to continue");

    receipt.setVisible(false);

}

JOptionPane.showMessageDialog(null, "All orders have been successfully
removed");

//If there are no orders, then we need to reset the labels, receipt screen and
productsList

if(orderListDropDownMenu.getItemCount()==0){

    productDetailsTitle.setText("Unable to show product details for an order at the
moment, consider making an order.");

    orderSubtotalTitle.setText("Unable to calculate order subtotal.");

    orderSubtotalValueTitle.setText("0.00");

    orderSubtotalValueTitle.setVisible(false);

    orderListDropDownMenu.setVisible(false);

    productList.add("No orders have been made at the moment, please make an
order to continue and then select your product details");

    productList.removeAll();

    productList.removeAll();

    orderContentsTitle.setText("No orders have been made, please make an order.");

    orderContentsValueTitle.setVisible(false);

    idCount=o;
}
```

```
    }

}

else{
    JOptionPane.showMessageDialog(null, "Unable to remove all orders, no orders
have been made, please make an order to proceed");

}

/*



This method will remove all products in the currently selected order

*/
private void removeAllPipesBtnActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    //If there are orders, proceed

    if(manager.getOrderListSize()>0){

        //If the order has products then proceed

        if(manager.getOrder(orderListDropDownMenu.getSelectedIndex()).getPipeList().size()>0
        ){

            JOptionPane.showMessageDialog(null, String.format("All pipes have been
successfully removed from Order: %d", orderListDropDownMenu.getSelectedIndex()+1));



            //Looping to remove all producst from the selected order

            for (int i=0; i<productList.getItemCount(); i++){

manager.getOrder(orderListDropDownMenu.getSelectedIndex()).getPipeList().remove(0
;

```

```
    }

    //Ressting the required labels to the first orders

    productList.removeAll();

    productList.removeAll();

    productDetailsTitle.setText(String.format("Product details for order %d:",

orderListDropDownMenu.getSelectedIndex() + 1));

    orderSubtotalTitle.setText(String.format("Order %d subtotal: £",

orderListDropDownMenu.getSelectedIndex() + 1));

orderSubtotalValueTitle.setText(manager.getOrder(orderListDropDownMenu.getSelected

Index()).getOrderCost());



//Updating the products screen

refresh();

orderSubtotalValueTitle.setText(manager.getOrder(orderListDropDownMenu.getSelected

Index()).getOrderCost());

//Updating the receipt

checkReceipt();

}

else{

    JOptionPane.showMessageDialog(null, "Unable to remove all items, no products

have been found for this order. ");

}

else{
```

```
JOptionPane.showMessageDialog(null, "Unable to remove all products, no order  
have been made.");
```

```
}
```

```
}
```

```
/**
```

```
* @param args the command line arguments
```

```
*/
```

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

```
/* Set the Nimbus look and feel */
```

```
//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
```

```
/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and  
feel.
```

```
* For details see
```

```
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
```

```
*/
```

```
try {
```

```
    for (javax.swing.UIManager.LookAndFeelInfo info :  
        javax.swing.UIManager.getInstalledLookAndFeels()) {
```

```
        if ("Nimbus".equals(info.getName())) {
```

```
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
```

```
            break;
```

```
}
```

```
    }

} catch (ClassNotFoundException ex) {

    java.util.logging.Logger.getLogger(Interface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

    java.util.logging.Logger.getLogger(Interface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

    java.util.logging.Logger.getLogger(Interface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

    java.util.logging.Logger.getLogger(Interface.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/* Create and display the form */

}

// Variables declaration - do not modify

private javax.swing.JLabel additionalOptionsItem1;

private javax.swing.JLabel additionalOptionsItem2;

private javax.swing.JLabel additionalOptionsTitle;
```

```
private javax.swing.JComboBox<String> chemicalResistanceDropDownMenu;
private javax.swing.JLabel chemicalResistanceTitle;
private javax.swing.JLabel colorPrintTitle;
private javax.swing.JComboBox<String> colorPrintValueDropDownMenu;
private javax.swing.JLabel companuTitler;
private javax.swing.JButton createOrderBtn;
private javax.swing.JLabel extrasPricingItem1;
private javax.swing.JLabel extrasPricingItem2;
private javax.swing.JLabel extrasPricingItem3;
private javax.swing.JLabel extrasPricingItem4;
private javax.swing.JLabel extrasPricingItem5;
private javax.swing.JLabel extrasPricingTitle;
private javax.swing.JButton getQuoteBtn;
private javax.swing.JButton getReceiptBtn;
private javax.swing.JLabel innerInsulatioTitle;
private javax.swing.JComboBox<String> innerInsulationValueDropDownMenu;
private javax.swing.JDialog jDialog1;
private javax.swing.JTextField lengthTextField;
private javax.swing.JLabel lengthTitle;
private javax.swing.JLabel orderContentsTitle;
private javax.swing.JLabel orderContentsValueTitle;
private javax.swing.JComboBox<String> orderListDropDownMenu;
private javax.swing.JLabel orderSubtotalTitle;
private javax.swing.JLabel orderSubtotalValueTitle;
private javax.swing.JTextField outerDiameterTextField;
```

```
private javax.swing.JLabel outerDiameterTitle;  
private javax.swing.JComboBox<String> outerReinforcementDropDownMenu;  
private javax.swing.JLabel outerReinforcementTitle;  
private javax.swing.JLabel plasticGradeTitle;  
private javax.swing.JComboBox<String> plasticGradeValueDropDownMenu;  
private javax.swing.JLabel productCostTitle;  
private javax.swing.JLabel productDetailsTitle;  
private java.awt.List productList;  
private java.awt.List productValueList;  
private javax.swing.JTextField qantityTextField;  
private javax.swing.JLabel qantityTitle;  
private javax.swing.JButton removeAllOrdersBtn;  
private javax.swing.JButton removeAllPipesBtn;  
private javax.swing.JButton removeOrderBtn;  
private javax.swing.JButton removePipeBtn;  
private javax.swing.JLabel selectPipeOptionsTitle;  
private javax.swing.JLabel usageItem1Title;  
private javax.swing.JLabel usageItem2Title;  
private javax.swing.JLabel usageItem3Title;  
private javax.swing.JLabel usageItem4Title;  
private javax.swing.JLabel usageTitle;  
// End of variables declaration  
}
```

Order Manag:

```
/*
```

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

```
package coursework;
```

```
import java.util.ArrayList;
```

```
/**
```

```
*
```

```
* @author up811216
```

```
*/
```

```
public class OrderManage {
```

```
    private int ID;
```

```
    private ArrayList<Order> orderList = new ArrayList<Order>();
```

```
    private double totalOrderCost=0;
```

```
/**
```

```
*Intitalsizing order Manag
```

```
* @param id
```

```
*/
```

```
public OrderManage(int id){
```

```
    ID = id;
```

```
}
```

```
/**  
 *Adding a new order  
 * @param ord  
 */  
  
public void createOrder(Order ord){  
    orderList.add(ord);  
}  
  
/**  
 *Removing an order  
 * @param index  
 */  
  
public void RemoveOrder(int index){  
    orderList.remove(index);  
}  
  
/**  
 * Retreiving an order based on index  
 * @param index  
 * @return  
 */  
  
public Order getOrder(int index){  
    return orderList.get(index);  
}
```

```
/**  
 *Retrieving the size of the orderList  
 * @return  
 */  
  
public int getOrderListSize(){  
    return orderList.size();  
}  
  
/**  
 *Receive the total cost for the order  
 * @return  
 */  
  
public String getOrderTotalCost(){  
    double temp=0;  
    for (Order i : orderList){  
        temp += Double.parseDouble(i.getOrderCost());  
        totalOrderCost = temp;  
    }  
  
    return String.format("%4.2f", totalOrderCost);  
}  
  
/**  
 *Retrieve the orderList  
 * @return
```

```
*/  
  
public ArrayList<Order> getOrder(){  
    return orderList;  
}  
}  
  
Order:  
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package coursework;  
  
import java.util.*;  
  
/**  
 *  
 * @author up811216  
 */  
  
public class Order {  
  
    private ArrayList<Pipe> itemList = new ArrayList<Pipe>();  
    private double volumeTotalCost = 0;  
    private double totalCost = 0;  
    private double orderCost = 0;  
    private double colNo1Cost = 0.12;  
    private double colNo2Cost = 0.16;
```

```
private double insulationCost = 0.13;  
private double resistCost = 0.14;  
private double reinforcedCost = 0.17;  
private double grade1Cost = 0.4;  
private double grade2Cost = 0.6;  
private double grade3Cost = 0.75;  
private double grade4Cost = 0.8;  
private double grade5Cost = 0.95;  
private double lengthInches = 39.37;  
private double lengthDiameter = 0.0254;  
private int ID;  
  
/**  
 *  
 * @param id  
 */  
public Order(int id){  
    ID = id;  
}  
  
/**  
 * Returns an array of Pipes  
 * @return  
 */
```

```
public ArrayList<Pipe> getPipeList(){
    return itemList;
}

/**
 *Returns the size of the PipeList Array
 * @return
 */

public int getPipeItemListSize(){
    return itemList.size();
}

}

/**
 *Returns the ID
 * @return
 */

public int getID(){
    return ID;
}

}

/**
 *Returns the first Color Cost Number
 * @return
 */

public double getColNo1Cost(){
```

```
    return colNo1Cost;  
}  
  
/**  
 *Returns the second color cost Number2  
 * @return  
 */  
public double getColNo2Cost(){  
    return colNo2Cost;  
}  
  
/**  
 *Changes the ID  
 * @param id  
 */  
public void changeID(int id){  
    ID = id;  
}  
  
/**  
 *Returns the insulation cost  
 * @return  
 */  
public double getInsulationCost(){  
    return insulationCost;
```

```
}

/***
 *Returns the resist cost
 * @return
 */
public double getResistCost(){
    return resistCost;
}

/***
 *Returns the Reinforced Cost
 * @return
 */
public double getReinforcedCost(){
    return reinforcedCost;
}

/***
 * Returns the grade 1 cost
 *
 * @return
 */
public double getGrade1Cost(){
    return grade1Cost;
}
```

}

/\*\*

\*Returns grade 2 cost

\* @return

\*/

public double getGrade2Cost(){

    return grade2Cost;

}

/\*\*

\*Returns the grade 3 cost

\* @return

\*/

public double getGrade3Cost(){

    return grade3Cost;

}

/\*\*

\*Returns the grade 4 cost

\* @return

\*/

public double getGrade4Cost(){

    return grade4Cost;

}

```
/**  
 *Returns the grade 5 cost  
 * @return  
 */  
  
public double getGrade5Cost(){  
    return grade5Cost;  
}  
  
/**  
 *Returns the volume total  
 * @return  
 */  
  
public double getVolumeTotal(){  
    return volumeTotalCost;  
}  
  
/**  
 *Returns the length compared to inches  
 * @return  
 */  
  
public double getLengthInches(){  
    return lengthInches;  
}
```

```
/**  
 *Returns the diameter length value  
 * @return  
 */  
  
public double getLengthDiameter(){  
    return lengthDiameter;  
}  
  
/**  
 *Returns the total cost  
 * @return  
 */  
  
public double getTotalCost(){  
    return totalCost;  
}  
  
/**  
 *Calculates the total cost  
 */  
  
public void calculateTotal(){  
    for(Pipe i : itemList){  
        totalCost += i.getPipeCost();  
    }  
}
```

```
/**  
 *Calculates the total volume  
 */  
  
public void calculateVolumeTotal(){  
    for(Pipe i : itemList){  
        totalCost += i.getVolume();  
    }  
}  
  
/**  
 *Adds a pipe to the item list  
 * @param p  
 */  
  
public void addPipe(Pipe p){  
    itemList.add(p);  
}  
  
/**  
 *Removes a pipe from the item  
 * @param p  
 */  
  
public void removePipe(Pipe p){  
    itemList.remove(p);  
}
```

```
/**  
 *Retrieving the pipe information  
 */  
  
public void getPipeInfoSimple(){  
    for (Pipe p : itemList){  
        p.getPipeInfoSimplified();  
    }  
}  
  
/**  
 *Retreiving the order cost  
 * @return  
 */  
  
public String getOrderCost(){  
    double temp=o;  
    if(itemList.size()>=1){  
        for (Pipe i : itemList){  
            temp += i.getPipeCost();  
        orderCost = temp;  
    }  
    }  
    else{  
        orderCost = o;  
    }  
}
```

```
    }  
  
    return String.format("%4.2f", orderCost);  
}  
  
}
```

Pipe:

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package coursework;  
  
import java.text.DecimalFormat;  
  
import java.util.*;  
  
/**  
 *  
 * @author up811216  
 */  
  
public abstract class Pipe {
```

```
/**  
 *  
 */  
  
protected double length = 0;
```

```
/**
```

\*

\*/

protected double outerDiameter = o;

/\*\*

\*

\*/

protected double volume = o;

/\*\*

\*

\*/

protected int quantity = o;

/\*\*

\*

\*/

protected int[] grade = new int[4];

/\*\*

\*

\*/

protected boolean chemicalResist = false;

/\*\*

\*

\*/

protected Order order;

/\*\*

\*

\*/

protected double pipeCost = o;

/\*\*

\*

\*/

protected double innerDiameter = o;

/\*\*

\*

\*/

protected double innerDiameterVolume = o;

/\*\*

\*

\*/

protected int gradeValue;

/\*\*

\*

\*/

protected double standardCost = o;

/\*\*

\*

\*/

protected String resist;

/\*\*

\*

\*/

protected double baseCost=o;

/\*\*

\*

\*/

protected DecimalFormat df = new DecimalFormat("#.oo");

/\*\*

\*

\* @param Length

\* @param OuterDiameter

\* @param Qantity

\* @param isChemicalResist

```
* @param GradeValue  
* @param ord  
*/  
  
public Pipe(double Length, double OuterDiameter, int Qantity, boolean  
isChemicalResist, int GradeValue, Order ord){  
  
    for(int i= 1; i<=3; i++){  
  
        grade[i-1] = i;  
    }  
  
    length = Length;  
    outerDiameter = OuterDiameter;  
    qantity = Qantity;  
    chemicalResist = isChemicalResist;  
    innerDiameter = 0.9*outerDiameter;  
    order = ord;  
  
    gradeValue = GradeValue;  
    CalculateInnerDiamaterVolume();  
    CalculateVolume();  
    standardCost = volume*order.getGrade1Cost();  
    calculatePipeCost();  
  
    if(chemicalResist){  
        resist = "Yes";  
    }  
  
    else{  
        resist = "No";  
    }  
}
```

}

abstract double getLength();

abstract double getOuterDiameter();

abstract double getVolume();

abstract double getQantity();

abstract boolean getChemicalResist();

abstract double getPipeCost();

abstract String getPipeCostStr();

abstract void setLength(double Length);

abstract void setOuterDiameter(double OuterDiameter);

abstract void setQantity(int Qantity);

abstract void setChemicalResist(boolean ChemicalResist);

```
abstract void CalculateVolume();

abstract void CalculateInnerDiamaterVolume();

abstract String getPipeInfoSimplified();

abstract void calculatePipeCost();

}
```

Type 1:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package coursework;
```

```
/***
 *
 * @author up811216
 */
```

```
public class Type1 extends Pipe{
```

```
/**  
*  
* @param Length  
* @param OuterDiameter  
* @param Qantity  
* @param isChemicalResist  
* @param GradeValue  
* @param ord  
*/  
  
public Type1(double Length, double OuterDiameter, int Qantity, boolean  
isChemicalResist, int GradeValue, Order ord){  
  
    super(Length, OuterDiameter, Qantity, isChemicalResist, GradeValue, ord);  
  
    for(int i= 1; i<=3; i++){  
  
        grade[i] = i;  
    }  
  
}  
  
/**  
*Retrieve Pipe Cost  
* @return  
*/  
  
public double getPipeCost(){  
  
    return pipeCost;  
}
```

```
/**  
 *Retrieve Pipe Information  
 * @return  
 */  
  
public String getPipeInfoSimplified(){  
  
    String info = String.format("Pipe grade: %d, Length: %4.3f, Outer diameter: %4.1f,  
    Qantity: %d, Chemical resistance: %s, Colours: No", gradeValue, length, outerDiameter,  
    quantity, resist);  
  
    return info ;  
}  
  
/**  
 *Retrieve Pipe Length  
 * @return  
 */  
  
public double getLength(){  
  
    return length;  
}  
  
/**  
 *Retrieve Pipe Outer Diameter  
 * @return  
 */  
  
public double getOuterDiameter(){  
  
    return outerDiameter;
```

}

/\*\*

\*Retrieve Pipe Volume

\* @return

\*/

public double getVolume(){

    return volume;

}

/\*\*

\*Retrieve Qantity

\* @return

\*/

public double getQantity(){

    return qantity;

}

/\*\*

\*Retrieve Pipe Chemical Resistance value

\* @return

\*/

public boolean getChemicalResist(){

    return chemicalResist;

}

```
/**  
 *Retrieve Pipe Cost in String format  
 * @return  
 */  
  
public String getPipeCostStr(){  
    String cost = String.format(df.format(pipeCost));  
    return cost;  
}  
  
/**  
 *Setting the Pipe Length  
 * @param Length  
 */  
  
public void setLength(double Length){  
    length = Length;  
}  
  
/**  
 *Setting the outer diameter  
 * @param OuterDiameter  
 */  
  
public void setOuterDiameter(double OuterDiameter){  
    outerDiameter = OuterDiameter;  
}
```

```
/**  
 *Setting the Pipe Qantity  
 * @param Qantity  
 */  
  
public void setQantity(int Qantity){  
    qantity = Qantity;  
}  
  
/**  
 *Setting the Chemical Resistance  
 * @param ChemicalResist  
 */  
  
public void setChemicalResist(boolean ChemicalResist){  
    chemicalResist = chemicalResist;  
}  
  
/**  
 *Calculating Pipe Volume  
 */  
  
public void CalculateVolume(){  
    volume = (Math.PI*(Math.pow(((outerDiameter)/2),  
    2))*(length*order.getLengthInches())) -innerDiameterVolume;  
}
```

```
/**  
 *Calculate inner Diameter Volume  
 */  
  
public void CalculateInnerDiamaterVolume(){  
    innerDiameterVolume = Math.PI*((Math.pow(((innerDiameter)/2),  
2))*(length*order.getLengthInches()));  
}  
  
//Calculating Pipe Cost  
  
/**  
 *Calculate the Pipe Cost  
 */  
  
public void calculatePipeCost(){  
    System.out.println(volume);  
    if(gradeValue==1){  
        pipeCost =  
Double.parseDouble(df.format(volume*(order.getGrade1Cost())))*qantity;  
    }else if(gradeValue==2){  
        pipeCost =  
Double.parseDouble(df.format(volume*(order.getGrade2Cost())))*qantity;  
    }  
    else{  
        pipeCost =  
Double.parseDouble(df.format(volume*(order.getGrade3Cost())))*qantity;  
    }  
    if(chemicalResist){
```

```
    pipeCost = (pipeCost*(order.getResistCost()+1));  
}  
}  
}
```

Type 2:

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package coursework;
```

```
/**
```

```
*
```

```
* @author up811216
```

```
*/
```

```
public class Type2 extends Type1{
```

```
    private int noCol = 1;
```

```
/**
```

```
*
```

```
* @param Length
```

```
* @param OuterDiameter
* @param Qantity
* @param isChemicalResist
* @param GradeValue
* @param ord
*/
public Type2(double Length, double OuterDiameter, int Qantity, boolean
isChemicalResist, int GradeValue, Order ord){
    super(Length, OuterDiameter, Qantity, isChemicalResist, GradeValue, ord);
    for(int i= 2; i<=4; i++){
        grade[i-2] = i;
    }
    calculatePipeCost();
}

/**
*Calculate the Pipe Cost
*/
public void calculatePipeCost(){
    if(gradeValue==2){
        pipeCost =
Double.parseDouble(df.format(volume*order.getGrade2Cost()*(order.getColNo1Cost()+1))
)*quantity;
    }
    else if (gradeValue==3){
```

```
    pipeCost =  
Double.parseDouble(df.format(volume*order.getGrade3Cost()*(order.getColNo1Cost()+1))  
)*quantity;  
}  
  
else{  
  
    pipeCost =  
Double.parseDouble(df.format(volume*order.getGrade4Cost()*(order.getColNo1Cost()+1))  
)*quantity;  
}  
  
  
if(chemicalResist){  
  
    double temp = pipeCost;  
  
    pipeCost = pipeCost+((temp*(order.getResistCost()+1))-temp);  
}  
  
}  
  
/**  
 *Retrieve Color value  
 * @return  
 */  
  
public int getColValue(){  
  
    return noCol;  
}  
  
  
//Retreiving the informaton fo this pipe  
  
  
/**Retrieve Pipe Information
```

```
*  
* @return  
*/  
  
public String getPipeInfoSimplified(){  
  
    String info = String.format("Pipe grade: %d, Length: %4.2f, Outer diameter: %4.1f,  
    Qantity: %d, Chemical resistance: %s, Colours: %d", gradeValue, length, outerDiameter,  
    quantity, resist, noCol);  
  
    return info ;  
}  
}
```

Type 3:

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

package coursework;

```
/**  
*  
* @author up811216  
*/
```

public class Type3 extends Type1 {

```
/**
```

```
*
```

\*/

protected int noCols = 2;

/\*\*

\*

\* @param Length

\* @param OuterDiameter

\* @param Qantity

\* @param isChemicalResist

\* @param GradeValue

\* @param ord

\*/

public Type3(double Length, double OuterDiameter, int Qantity, boolean isChemicalResist, int GradeValue, Order ord){

super(Length, OuterDiameter, Qantity, isChemicalResist, GradeValue, ord);

for(int i= 2; i<=5; i++){

grade[i-2] = i;

}

calculatePipeCost();

}

/\*\*

\*Retrieve the pipe color

\* @return

\*/

public int getNoCols(){

```
    return noCols;  
}  
  
/**  
 *Retrieve Pipe Information  
 * @return  
 */  
public String getPipeInfoSimplified(){  
    String info = String.format("Pipe grade: %d, Length: %4.2f, Outer diameter: %4.1f,  
    Qantity: %d, Chemical resistance: %s, Colours: %d", gradeValue, length, outerDiameter,  
    quantity, resist, noCols);  
    return info ;  
}  
  
/**  
 *Calculate Pipe. This method will calculate the pie cost based on the volume of the  
 pipe, the grade cost and the color cost  
 */  
public void calculatePipeCost(){  
    if(gradeValue==2){  
        pipeCost =  
Double.parseDouble(df.format((volume*order.getGrade2Cost())*(order.getColNo2Cost()  
+  
1)))*quantity;  
    }  
    else if (gradeValue==3){  
        pipeCost =  
Double.parseDouble(df.format((volume*order.getGrade3Cost())*(order.getColNo2Cost()  
+  
1)))*quantity;  
    }  
}
```

```
    }

    else if (gradeValue==4){

        pipeCost =
Double.parseDouble(df.format((volume*order.getGrade4Cost())*(order.getColNo2Cost()+
1)))*quantity;

    }

    else if(gradeValue == 5){

        pipeCost =
Double.parseDouble(df.format((volume*order.getGrade5Cost())*(order.getColNo2Cost()+
1)))*quantity;

    }

    if(chemicalResist){

        double temp = pipeCost;

        pipeCost = pipeCost+((temp*(order.getResistCost()+1))-temp);

    }

}

Type 4:

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package coursework;
```

```
/**  
*  
* @author up811216  
*/  
public class Type4 extends Type3{  
  
    /**  
     *  
     */  
    protected boolean insulation = true;  
  
    /**  
     *  
     * @param Length  
     * @param OuterDiameter  
     * @param Qantity  
     * @param isChemicalResist  
     * @param GradeValue  
     * @param ord  
     */  
    public Type4(double Length, double OuterDiameter, int Qantity, boolean  
isChemicalResist, int GradeValue, Order ord){  
        super(Length, OuterDiameter, Qantity, isChemicalResist, GradeValue, ord);  
        calculatePipeCost();  
    }
```

```
/*
 * Retrieve Pipe Information
 */
public String getPipeInfoSimplified(){
    String info = String.format("Pipe grade: %d, Length: %4.2f, Outer diameter: %4.1f,
    Qantity: %d, Chemical resistance: %s, Colours: %d, Insulated: Yes",
        gradeValue, length, outerDiameter, quantity, resist, noCols);

    return info ;
}

/**
 *Retrieve Pipe Insulation value
 * @return
 */
public boolean getInsultaion(){

    return insulation;
}

/*
Calculate Pipe cost based on inner insulatio, color value, and grade cost
*/
public void calculatePipeCost(){

    if(gradeValue==2){

        pipeCost =
Double.parseDouble(df.format((volume*order.getGrade2Cost())*((order.getColNo2Cost()
+order.getInsulationCost())+1)))*quantity;
    }
}
```

```
else if (gradeValue==3){  
    pipeCost =  
Double.parseDouble(df.format((volume*order.getGrade3Cost())*((order.getColNo2Cost()  
+order.getInsulationCost())+1)))*quantity;  
}  
  
else if (gradeValue==4){  
    pipeCost =  
Double.parseDouble(df.format((volume*order.getGrade4Cost())*((order.getColNo2Cost()  
+order.getInsulationCost())+1)))*quantity;  
}  
  
else if(gradeValue == 5){  
    pipeCost =  
Double.parseDouble(df.format((volume*order.getGrade5Cost())*((order.getColNo2Cost()  
+order.getInsulationCost())+1)))*quantity;  
}  
  
if(chemicalResist){  
    double temp = pipeCost;  
    pipeCost = pipeCost+((temp*(order.getResistCost())+1))-temp);  
}  
}  
}  
}
```

Type 5:

```
/*
```

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

```
*/
```

package coursework;

```
/**
```

```
*
```

```
* @author up811216
```

```
*/
```

```
public class Type5 extends Type4{
```

```
    private boolean reInforcement = true;
```

```
    /**
```

```
*
```

```
* @param Length
```

```
* @param OuterDiameter
```

```
* @param Qantity
```

```
* @param isChemicalResist
```

```
* @param GradeValue
```

```
* @param ord
```

```
*/
```

```
    public Type5(double Length, double OuterDiameter, int Qantity, boolean  
isChemicalResist, int GradeValue, Order ord){
```

```
        super(Length, OuterDiameter, Qantity, isChemicalResist, GradeValue, ord);
```

```
        for(int i= 3; i<=5; i++){
```

```
            grade[i-3] = i;
```

```
}
```

```
        calculatePipeCost();
```

```
}
```

```
/**
```

```
*Retrieve Pipe Reinforcement value
```

```
* @return
```

```
*/
```

```
public boolean getReinforcement(){
```

```
    return reInforcement;
```

```
}
```

```
/*
```

```
Retrieve Pipe Information
```

```
*/
```

```
public String getPipeInfoSimplified(){
```

```
    String info = String.format("Pipe grade: %d, Length: %4.2f, Outer diameter: %4.1f,  
    Qantity: %d, Chemical resistance: %s, Colours: %d, Insulated: Yes, Reinforced: Yes",  
    gradeValue, length, outerDiameter, qantity, resist, noCols);
```

```
    return info ;
```

```
}
```

```
/*
```

```
* CalculatePipeCost - This method will use the insulation cost, the color value cost, and  
grade cost and outer reinforcement values to produce a pipeCost
```

```
*/
```

```
public void calculatePipeCost(){
```

```
if(gradeValue==3){  
    pipeCost =  
Double.parseDouble(df.format((volume*order.getGrade3Cost())*((order.getColNo2Cost()  
+order.getInsulationCost()+order.getReinforcedCost())+1)))*qantity;  
}  
  
else if (gradeValue==4){  
    pipeCost =  
Double.parseDouble(df.format((volume*order.getGrade4Cost())*((order.getColNo2Cost()  
+order.getInsulationCost()+order.getReinforcedCost())+1)))*qantity;  
}  
  
else if (gradeValue==5){  
    pipeCost =  
Double.parseDouble(df.format((volume*order.getGrade5Cost())*((order.getColNo2Cost()  
+order.getInsulationCost()+order.getReinforcedCost())+1)))*qantity;  
}  
  
if(chemicalResist){  
    double temp = pipeCost;  
    pipeCost = pipeCost+((temp*(order.getResistCost())+1))-temp);  
}  
  
}  
}
```

Receipt:

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

```
*/  
package coursework;  
  
import java.awt.event.WindowEvent;  
  
/**  
 *  
 * @author up811216  
 */  
public class Receipt extends javax.swing.JFrame {  
  
    /**  
     * Creates new form Receipt  
     */  
  
    private Main Main;  
    private OrderManage manager;  
    private int numItems = 0;  
    public Receipt(OrderManage orderManager) {  
        initComponents();  
        Main = Main;  
        manager = orderManager;  
        populateComponents();  
    }  
}
```

```
/**  
 * This method is called from within the constructor to the form.  
 * WARNING: Do NOT modify this code. The content of this method is always  
 * regenerated by the Form Editor.  
 */  
  
@SuppressWarnings("unchecked")  
// <editor-fold defaultstate="collapsed" desc="Generated Code">  
private void initComponents() {  
  
    companyTitle2 = new javax.swing.JLabel();  
    totalCostTitle = new javax.swing.JLabel();  
    totalCostValueTitle = new javax.swing.JLabel();  
    orderDetailsTitle = new javax.swing.JLabel();  
    thankYouMessageTitle = new javax.swing.JLabel();  
    productsList = new java.awt.List();  
    totalCostValueCurrencyTitle = new javax.swing.JLabel();  
    orderDetailsTitle1 = new javax.swing.JLabel();  
    orderDetailsTitle2 = new javax.swing.JLabel();  
    itemsOrderedTitle = new javax.swing.JLabel();  
    itemsOrderedValue = new javax.swing.JLabel();  
    productsValueList = new java.awt.List();  
    productValueTilte = new javax.swing.JLabel();  
    orderLocationList = new java.awt.List();  
    orderLocationTitle = new javax.swing.JLabel();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

companyTitle2.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
companyTitle2.setText("Long Pipes");

totalCostTitle.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
totalCostTitle.setText("Total cost:");

totalCostValueTitle.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
totalCostValueTitle.setText("o.oo");

orderDetailsTitle.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
orderDetailsTitle.setText("Your order details:");

thankYouMessageTitle.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
thankYouMessageTitle.setText("Thank you for shopping at Long Pipes");

productsList.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        productsListActionPerformed(evt);
    }
});

totalCostValueCurrencyTitle.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```

```
totalCostValueCurrencyTitle.setText("£");
```

```
orderDetailsTitle1.setText("This is the products that you have ordered from all of your  
order that you have made.");
```

```
orderDetailsTitle2.setText("To make an amendment, please refer to the ordering  
screen.");
```

```
itemsOrderedTitle.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
itemsOrderedTitle.setText("#Items ordered:");
```

```
itemsOrderedValue.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
itemsOrderedValue.setText("0");
```

```
productValueTilte.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
productValueTilte.setText("Product cost: (£)");
```

```
orderLocationTitle.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
orderLocationTitle.setText("From:");
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());  
getContentPane().setLayout(layout);  
layout.setHorizontalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(layout.createSequentialGroup())
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(20, 20, 20)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(orderDetailsTitle2)
        .addComponent(orderDetailsTitle1)
        .addGroup(layout.createSequentialGroup()
            .addComponent(orderDetailsTitle)
            .addGap(163, 163, 163)
            .addComponent(companyTitle2))
        .addGroup(layout.createSequentialGroup()
            .addComponent(itemsOrderedTitle)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(itemsOrderedValue)
        .addGap(215, 215, 215)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(totalCostTitle)
        .addGroup(layout.createSequentialGroup()
            .addComponent(totalCostValueCurrencyTitle)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(totalCostValueTitle))))
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)

    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()

        .addComponent(orderLocationTitle)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(productValueTilte))

    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()

        .addComponent(orderLocationList,
javax.swing.GroupLayout.PREFERRED_SIZE, 115,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(productsList,
javax.swing.GroupLayout.PREFERRED_SIZE, 546,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(productsValueList,
javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE)))))

    .addGroup(layout.createSequentialGroup()

        .addGap(265, 265, 265)

        .addComponent(thankYouMessageTitle))

    .addContainerGap(17, Short.MAX_VALUE))

);
```

layout.setVerticalGroup(



```
.addComponent(productsList, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addComponent(orderLocationList,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 138,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(productsValueList,
javax.swing.GroupLayout.PREFERRED_SIZE, 136,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(3, 3, 3)

.addComponent(totalCostTitle)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(totalCostValueTitle)

.addComponent(totalCostValueCurrencyTitle)

.addComponent(itemsOrderedTitle)

.addComponent(itemsOrderedValue))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(thankYouMessageTitle)

.addContainerGap())

);

pack();

}// </editor-fold>
```

```
private void productsListActionPerformed(java.awt.event.ActionEvent evt) {
```

```
// TODO add your handling code here:  
}  
  
/**Updates the productsList, orderLocationList and Products Values List, calculates  
number of items  
 * @param args the command line arguments  
 */  
  
public void populateComponents(){  
    numItems=0;  
    productsList.removeAll();  
    productsValueList.removeAll();  
    orderLocationList.removeAll();  
    for(Order i : manager.getOrder()){  
        for (Pipe j : i.getPipeList()){  
            productsList.add(j.getPipeInfoSimplified());  
            productsValueList.add(j.getPipeCostStr());  
            orderLocationList.add("Order: "+(i.getID()+1));  
            numItems+=1;  
        }  
    }  
  
    totalCostValueTitle.setText(manager.getOrderTotalCost());  
    itemsOrderedValue.setText(""+ numItems);  
}
```

```
/*
```

```
Retrieve Number of Items
```

```
*/
```

```
public int getNumCount(){
```

```
    return numItems;
```

```
}
```

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

```
    /* Set the Nimbus look and feel */
```

```
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
```

```
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
```

```
     * For details see
```

```
     http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
```

```
    */
```

```
    try {
```

```
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
```

```
            if ("Nimbus".equals(info.getName())) {
```

```
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
```

```
                break;
```

```
        }

    }

} catch (ClassNotFoundException ex) {

    java.util.logging.Logger.getLogger(Receipt.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

    java.util.logging.Logger.getLogger(Receipt.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

    java.util.logging.Logger.getLogger(Receipt.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

    java.util.logging.Logger.getLogger(Receipt.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/* Create and display the form */

}

// Variables declaration - do not modify

private javax.swing.JLabel companyTitle2;
```

```
private javax.swing.JLabel itemsOrderedTitle;  
private javax.swing.JLabel itemsOrderedValue;  
private javax.swing.JLabel orderDetailsTitle;  
private javax.swing.JLabel orderDetailsTitle1;  
private javax.swing.JLabel orderDetailsTitle2;  
private java.awt.List orderLocationList;  
private javax.swing.JLabel orderLocationTitle;  
private javax.swing.JLabel productValueTilte;  
private java.awt.List productsList;  
private java.awt.List productsValueList;  
private javax.swing.JLabel thankYouMessageTitle;  
private javax.swing.JLabel totalCostTitle;  
private javax.swing.JLabel totalCostValueCurrencyTitle;  
private javax.swing.JLabel totalCostValueTitle;  
// End of variables declaration  
}
```

# Coursework

**Hand out: 23.X.2017    Submission (Moodle): 8.XII.2017 (Demonstration: by week11- starting 4th Dec)**

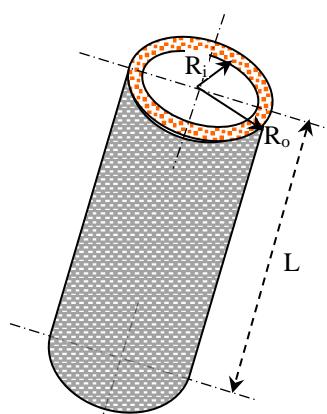
This is an assessed piece of group (of two) coursework, it is therefore essential to be completed and handed-in on time. If you are unclear about any aspect of the assignment, including the assessment criteria, please raise this at the first opportunity. The usual regulations apply to a late submission of work. The submitted application must be in Java (using Java NetBeans IDE) to be marked. During the demonstration (by week 11, in your lab session) you have to submit a memory stick with your source code and Java NetBeans project files with your group number on it.

The coursework you submit should be your group work. If your coursework includes other people's ideas and material, they must be properly referenced or acknowledged. Failing to do so intentionally or unintentionally constitutes plagiarism. The University treats plagiarism as a serious offence.

## ORDER SYSTEM FOR A PIPE-SELLING COMPANY

“*LongPipes*” is a company producing a variety of pipes for water, drainage, fuel, gas, conduit, plumbing and heating. Due to the wide range of requirements of their customers, the variety of pipes they produce is very extensive.

The pipes are all made of plastic, but some may have metallic enforcement and other features:



**Fig.1.** Pipe's cross-section.

- They are all made of plastic;
- Their plastic has a specified grade;
- They may have no colour, 1 colour, or 2 colours;
- They may have inner insulation layer;
- They may have outer metallic reinforcement;
- They may also have improved chemical resistance.

The types of pipes, produced by the company are shown in Table 1.  
**Table 1.** Types of plastic pipes available.

Type	Plastic's grade	Colour print			Inner insulation	Outer reinforcement	Chemical resistance
		0	1	2			
I	1 – 3	YES	NO	NO	NO	NO	YES/NO
II	2 – 4	NO	YES	NO	NO	NO	YES/NO
III	2 – 5	NO	NO	YES	NO	NO	YES/NO
IV	2 – 5	NO	NO	YES	YES	NO	YES/NO
V	3 – 5	NO	NO	YES	YES	YES	YES/NO

Pipes are available in straights (up to 6 meters). When ordering, the client should specify the length (in meters, 1m = 39.37") and the outer diameter (or outer radius  $R_o$ ) – in inches ( $1" = 0.0254\text{m}$ ). Assume the inner diameter (or the inner radius  $R_i$ ) is always 90% of the outer one. The basic cost is calculated using the volume of the pipe's material (in cubic inches) and the costs of 1 inch<sup>3</sup> of plastic is given in Table 2.

**Table 3.** Additional costs.

**Table 2.** Cost of 1 cubic inch of plastic

Grade of plastic	1	2	3	4	5
Cost per inch <sup>3</sup> [in £]	0.4	0.6	0.75	0.8	0.95

1 colour	12% extra
2 colours	16% extra
Inner insulation	13% extra
Outer reinforcement	17% extra
Chemical resistance	14% extra

There are some additional costs, depending on whether the pipe has colour and if there is any insulation and/or reinforcement. These are shown in Table 3 and the percentage increase is calculated using the basic cost.

All pipes may have improved chemical resistance. When customers ask *LongPipes* to quote a price for an order, they specify the following features:

- The size of pipe (length in meters and outer diameter in inches);
- The grade of the plastic;
- Whether they want any colour (no colour, or 1, or 2 colour plastic);
- Whether they want any insulation or/reinforcement;
- Whether they want pipe with chemical resistance;
- The quantity of pipes for the order.

From this information, the order system should determine if the requested type of pipe can be supplied by *LongPipes*, and if not, it should display an appropriate message and reject the order (e.g., pipe of plastic grade 1 and inner insulation is an invalid order). If the ordered pipe corresponds to one of the types given in Table 1, and can be supplied by *LongPipes*, the cost of the pipes must be calculated (using Table 2 and Table 3) and quoted.

The customer should be able to place several orders in one session, in which case the total cost should be prompted.

**Customers should not be asked for the type of pipe they want** (since this is only used within the company to calculate the cost).

**It is your application that must determine** (using Table 1) **the type of pipes based on the ordered pipe characteristics**.

Customers should be able to receive a quote for as many pipes (of different types) as they like (within the capacity of ***LongPipes***) in the same order. In such cases, the total cost of the order should be calculated and displayed.

Your user interface should be a GUI (graphical user interface) using AWT/Swing. If no GUI is used, you will lose the marks allocated for this part of your coursework.

## Your Task

- Write an application that allows the customer to enter the details of his/her order and subsequently prompts the cost of the order. Your application should verify that ***LongPipes*** can supply the corresponding to the order type of pipe (the customer should not be asked to specify the pipe type).
- Use OO design approach (abstraction, inheritance and polymorphism) and create appropriate class hierarchy, which reflects on the types of pipes that ***LongPipes*** sells. Use an abstract class as well.
- Give UML use case diagram, UML class hierarchy diagram, one class and one instance diagrams.
- Use proper level of abstraction, encapsulation and accessibility for the class attributes and methods. Application with no levels of abstraction will fail.
- Devise suitable test plan and data, which you can use to test the performance of your ordering system.

## Assessment Criteria

You should give **a demonstration and submit a memory stick** (with **your group number** on it) with your source code and Java NetBeans project files of your software no later than week11 (starting **4.XII.2017**), during your lab session.

On **8.XII.2017** your group should submit electronically (**by 6pm**) to Moodle a .pdf file with your **report**. **The file name should be your group name** (e.g., ***GrC-2.pdf***, or ***GrA-3.pdf***, or ***GrD-5.pdf***, etc.) **and should include** the following:

- ✓ **A UML** use case diagram of your order system, UML class hierarchy diagram of your OO application design, and also one UML class diagram (one class of your choice), and one instance diagram;
- ✓ **A brief** description of the application including any assumptions you have made and any limitations in your implementation of the application;
- ✓ **A test** schedule and screen shots to evidence the testing and evaluation;
- ✓ **The source code** that you have written as an Appendix (the same code that you used in your demonstration);
- ✓ **Some sample** input and output (screenshots) to demonstrate your application is working;
- ✓ **A Group contribution form** with your individual contributions;
- ✓ **This document**.

The assessment criteria and marks distribution are given in Table 4.

**Table 4.** Assessment criteria and marks distribution.

Topic/Criteria	Comments	Marks available	Marks awarded
Class hierarchy descriptions (UML)	How suitable is the design and the adopted hierarchy for the application? Use of abstract class?	10 (Report)	
UML class and instance diagrams	Are the UML use case, class and instance diagrams relevant to the application?	10 (Report)	
Code and functionality	How complete is the implementation? Does it perform as specified? Does it implement an OO design approach? Use of abstract class? Are the class attributes and methods at the appropriate hierarchy level? Is the verification and validation of input data adequate? Is the exception handling properly done? Are the style, indentation and comments appropriate? Is the layout clear?	45 (Demo(20), Report(25))	
Using AWT/Swing	How well designed is the interface? How appropriate is the use of components? How appropriate is the use of attributes? Is it working, or just an attempt? Is the layout clear?	15 (Demo)	
Testing	How thorough is planning and testing? Does it cover most/few possible errors?	10 (Report)	
Supporting documentation and comments.	Is the text clearly written and well presented? Are the assumptions, limitations, problems and features of the application well documented?	10 (Report)	
OVERALL MARK		100	

## **ADPROC Coursework - Group Contribution**

Complete the Group Members' Contribution to the ADPROC Coursework **below**.

This should cover the overall contribution of each group member to the coursework.

### **Group Members' Contribution to the ADPROC Coursework**

Distribute 100% among all the members of your group (including yourself) to indicate each person's relative contribution.

For example, in a group of four students Alpha, Beta, Gamma, and Delta, where all have contributed evenly, you would give 25% each.

However, if the contributions were uneven, you might mark them as e.g.: Alpha has done most of the work, so give her/him 50%; Beta, Gamma and Delta have completed the rest of the work and between them Beta did 25%, Gamma did 15% and Delta - 10%.

List your group number and group members **by student number** and their scores below:

Group No: GrD5

Student number and contribution:

- |                |           |
|----------------|-----------|
| 1. 811216_____ | 33.33/100 |
| 2. 812110_____ | 33.33/100 |
| 3. 811828_____ | 33.33/100 |

**TOTAL 100/100**