**FlexBox Ordering System**

**Description**

The application that I have designed is for a company called FlexBox which sell boxes of customizable sizes with addable features. The application was created using an Object Oriented Approach. The application allows you to pick the size constraints; length, height and width of the box, then the grade of the card being the type of card to be used for said box. Next is how many colours you would like the card to be; there is a choice from zero, one or two colours. Extra features are whether the box should have a sealable top, reinforced bottoms and reinforced corners. The company ‘FlexBox’ specifically create 5 types of boxes which the customers are unbeknownst to. Therefore when a box selection is made, not matching one of the five, an error message will show up informing them that type of box is not available/made by FlexBox.

Once a selection has been made which is one of the valid types of boxes allowed, it is added to an array list as a string and shows all the attributes/properties of the box as well as the total price of that single box order. Quantity is included in the price shown in the list. Multiple single orders can be added to the order (array list)

The java code uses an abstract class; BoxType and uses that to create 5 sub classes being the types of boxes which the company sell. There are abstract methods mentioned in the abstract class which the sub classes make use of showing abstraction in the code. Inheritance and polymorphism is present in the code too.

**Assumptions**

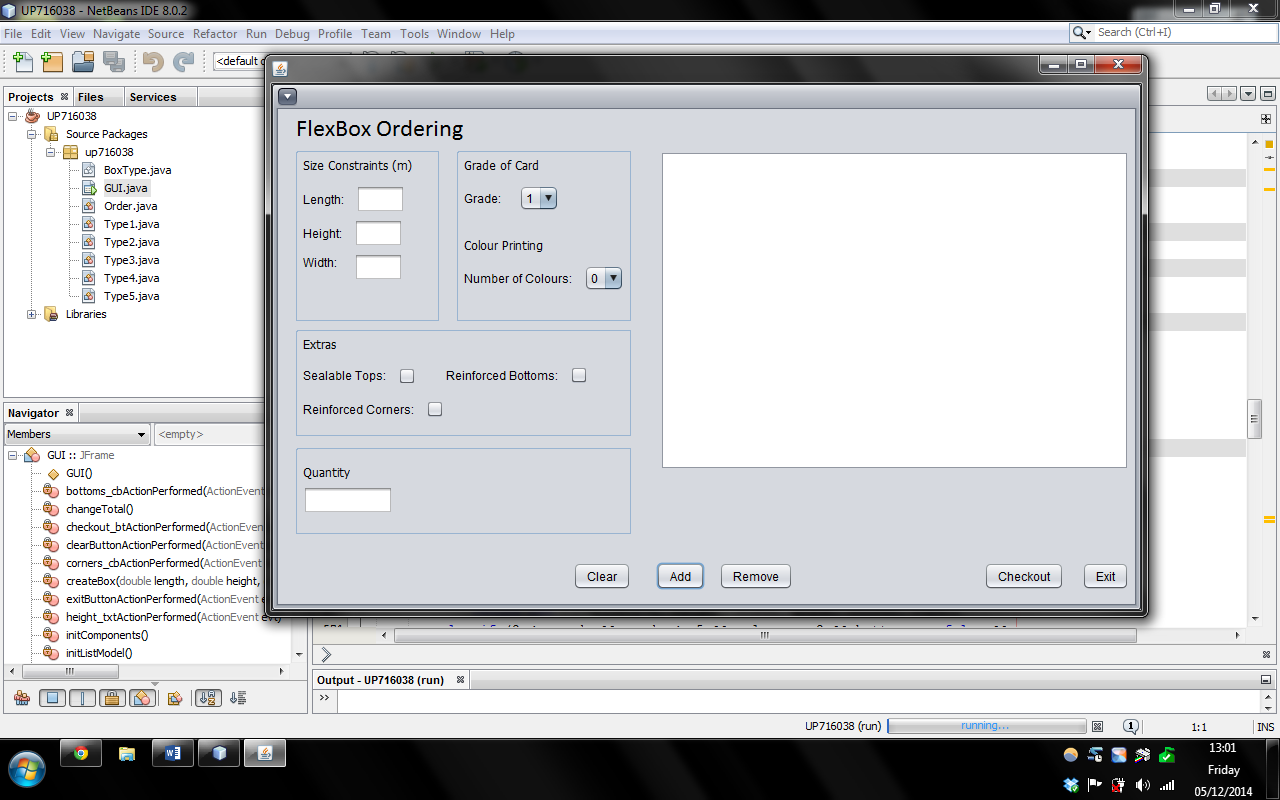
* The boxes provided by FlexBox have size constraints measured in metres.
* In order to place a single order, all text input fields have to be filled in being the size constraints and the quantity.
* A choice of grade of card and the number of colours are required fields which have a specific number of options that can be chosen and so will be introduced in combo boxes, allowing to pick different options through the use of drop down menus.
* The extra optional features being sealable tops, reinforced corners and reinforced bottoms will be implemented through the use of checkboxes.
* Every single order added will be represented as an object in an array list where all its attributes and the price (quantity included) will be shown in a JList.
* The user will be able to remove any unwanted orders from the Jlist through the use of a ‘remove’ button.
* The user will be able to clear all parameters defined in the input fields through the use of a ‘clear’ button.
* The user will be able to checkout with the wanted orders through the use of a ‘checkout’ button which once clicked will notify the user of a successful transaction and clear all the input text fields as well as the JList.

**Limitations**

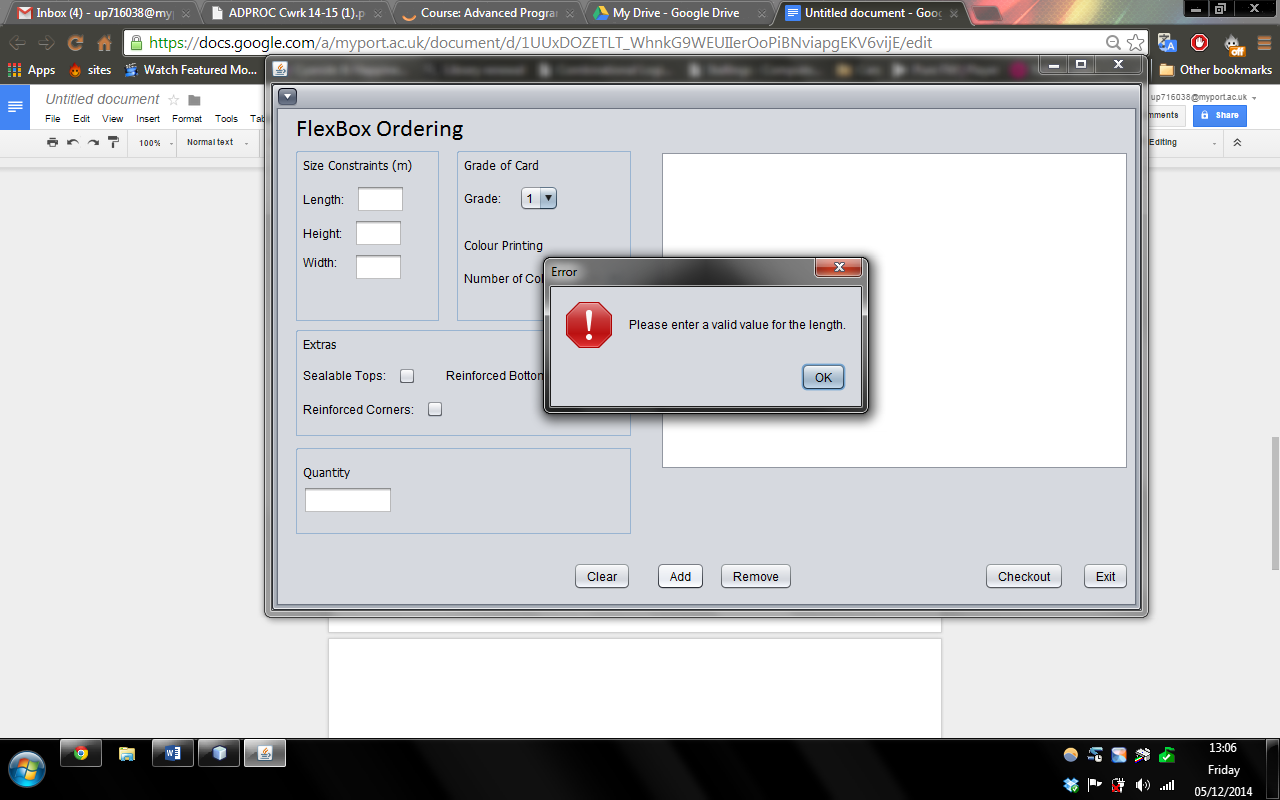
* There is a realistic range applied to the size constraints for the program.
  + Length has to be within 1 - 5 metres
  + Height has to be within 1 - 4 metres
  + Width has to be within 1 - 5 metres
* The grade of card can only be chosen from a range of 1 - 5.
* The number of colours can only be chosen from a range of 0 - 2.
* No characters other than numerical values can be put into the input text fields. 1f or 1d are exceptions to this rule. The program will take them in as their specified values in Java.
* Invalid characters entered into the input fields will not be accepted by the application hence displaying a message to the user of invalid characters in which fields. If there are multiple fields with invalid characters, the first field will have to be fixed and then checked and then will the application tell you which second field has an invalid character.
* There is a limit on the quantity of boxes that can be ordered for one single order/selection. A minimum of 1 and a maximum of 99 boxes of a single type can be ordered. However, if need be, the customer can order the same type once again at a limit of 99 boxes per single order.

**Testing**

1. When an order is added with empty parameters, a dialog box pops up informing the user to enter a valid value for the length:

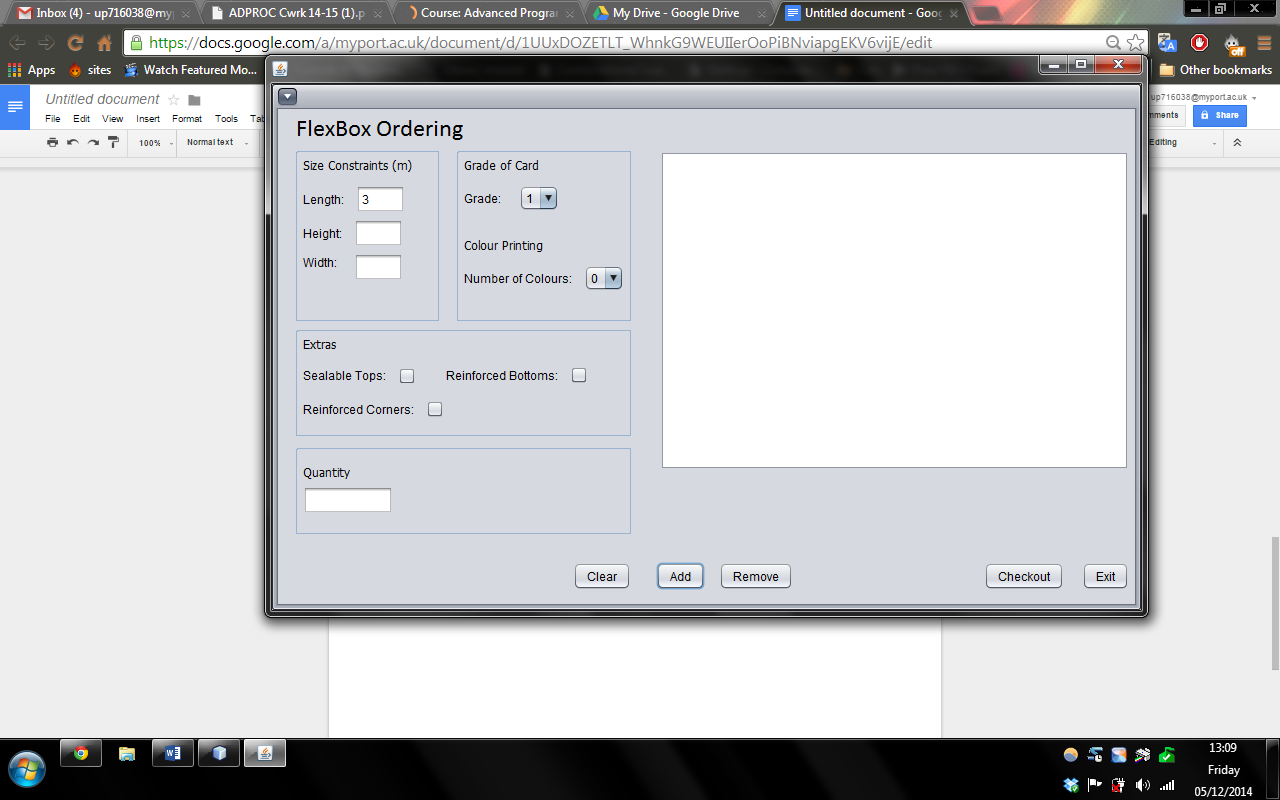


Empty Parameters

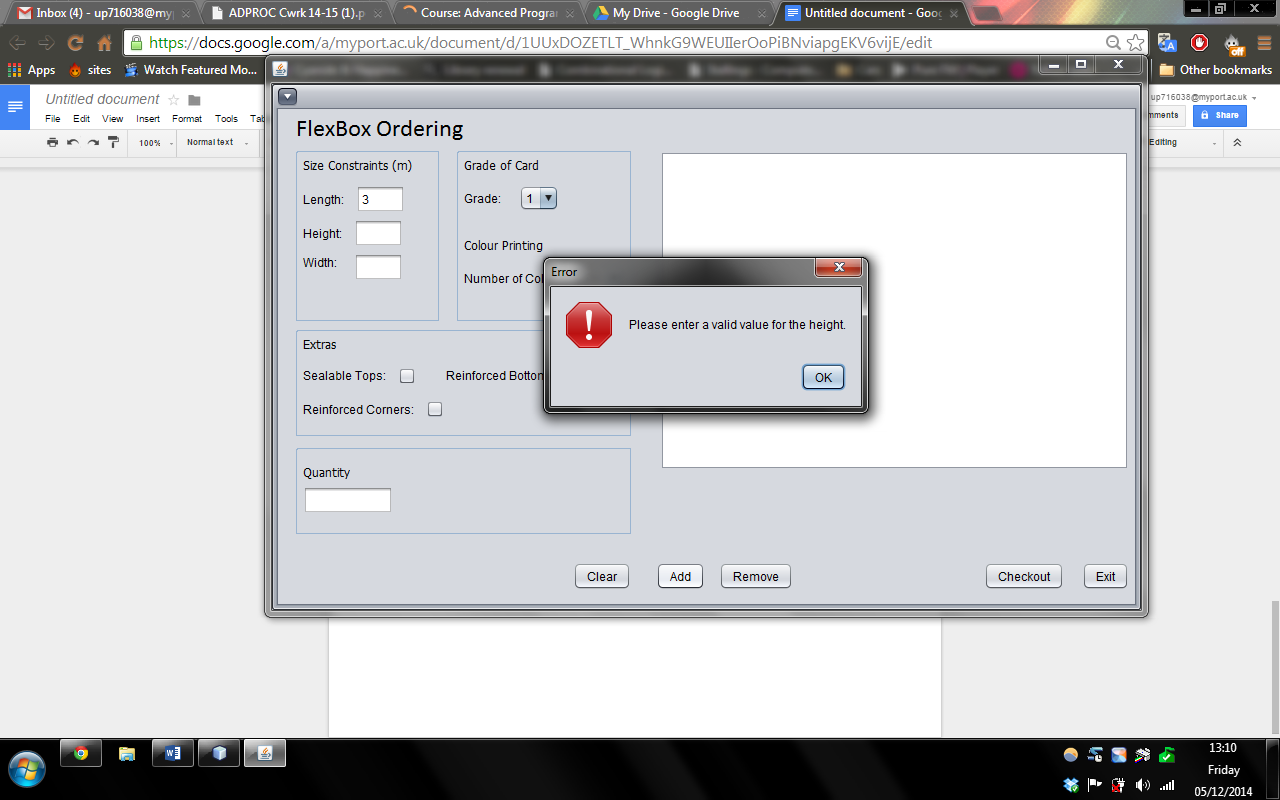


Error message informs the user about invalid value in the length text field.

1. When only the length text field has been entered in the parameters:

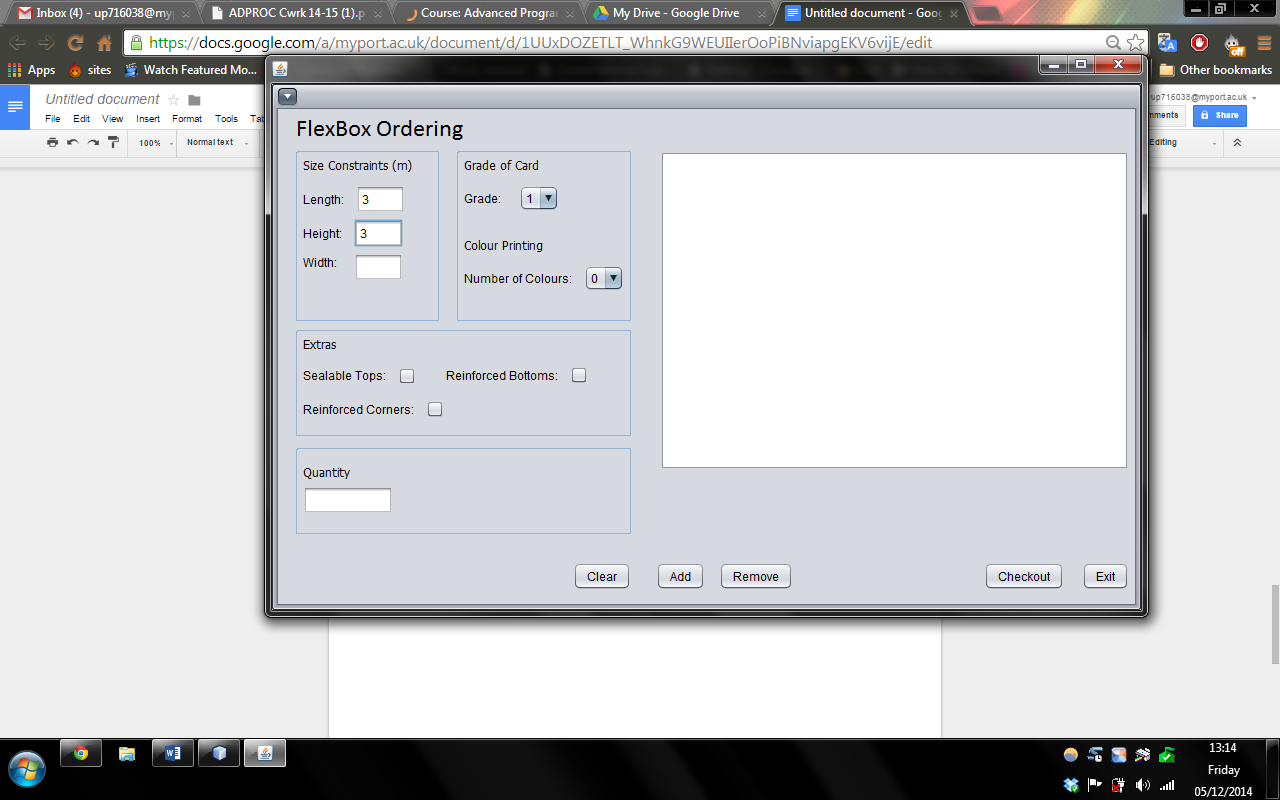


The length text field contains a value of 3.

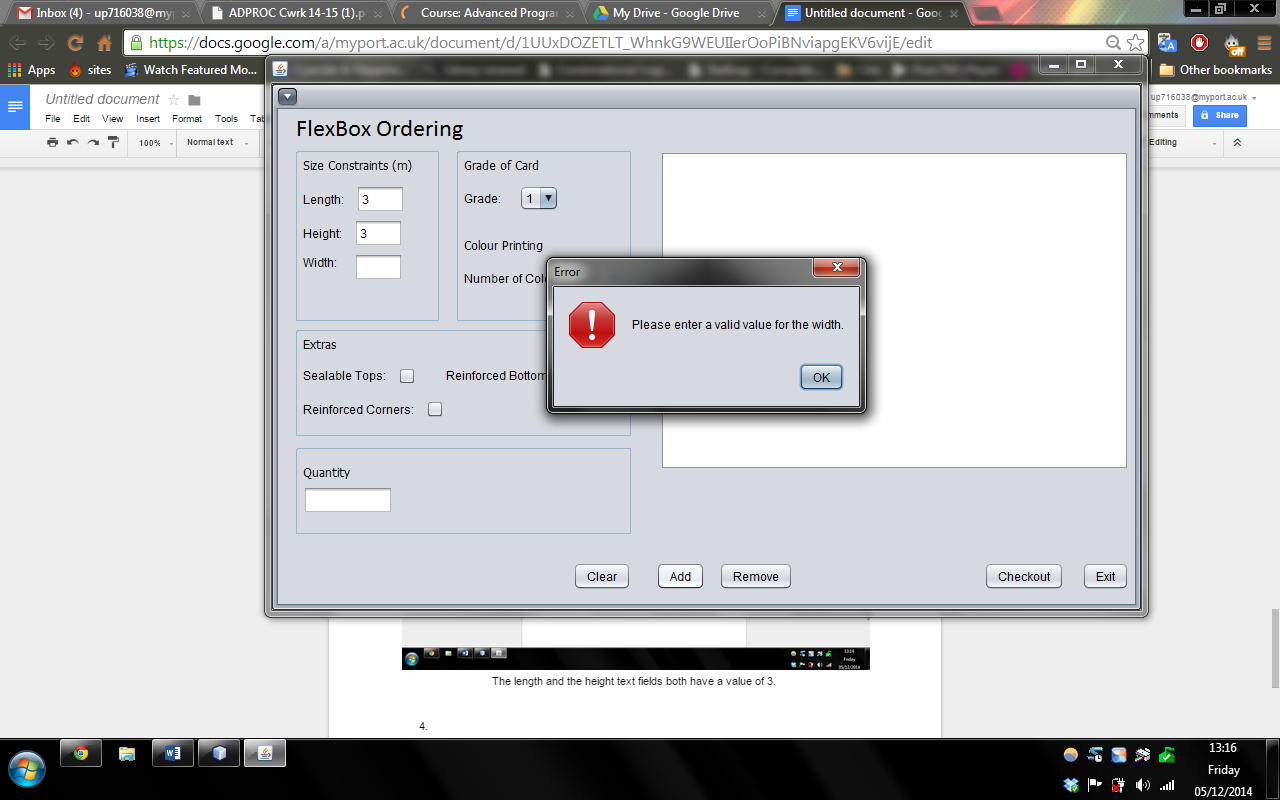


Error message informs the user about invalid value in the height text field.

1. When only the length and the height text fields are entered in the parameters and added as a single order:



The length and the height text fields both have a value of 3.

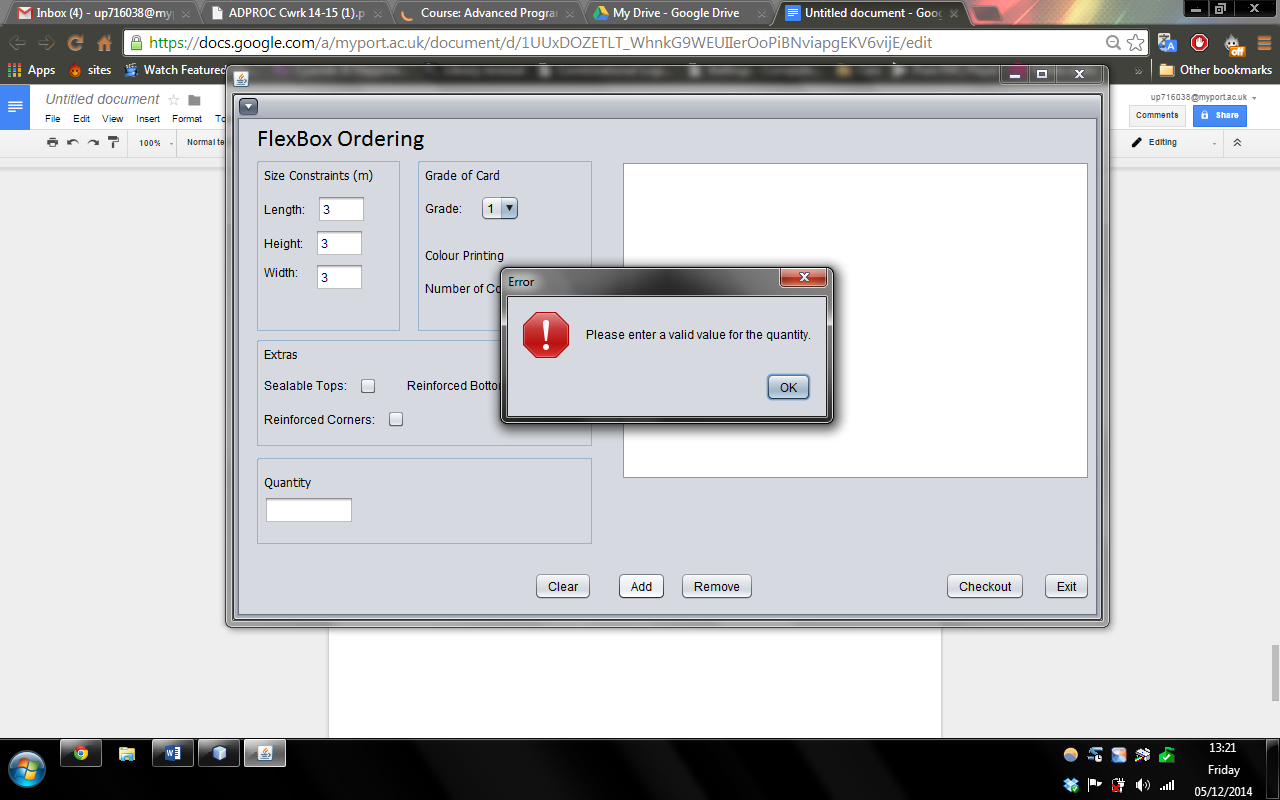


An error message informs the user of an invalid value in the width text field.

1. When all the text fields have been entered except for the quantity:

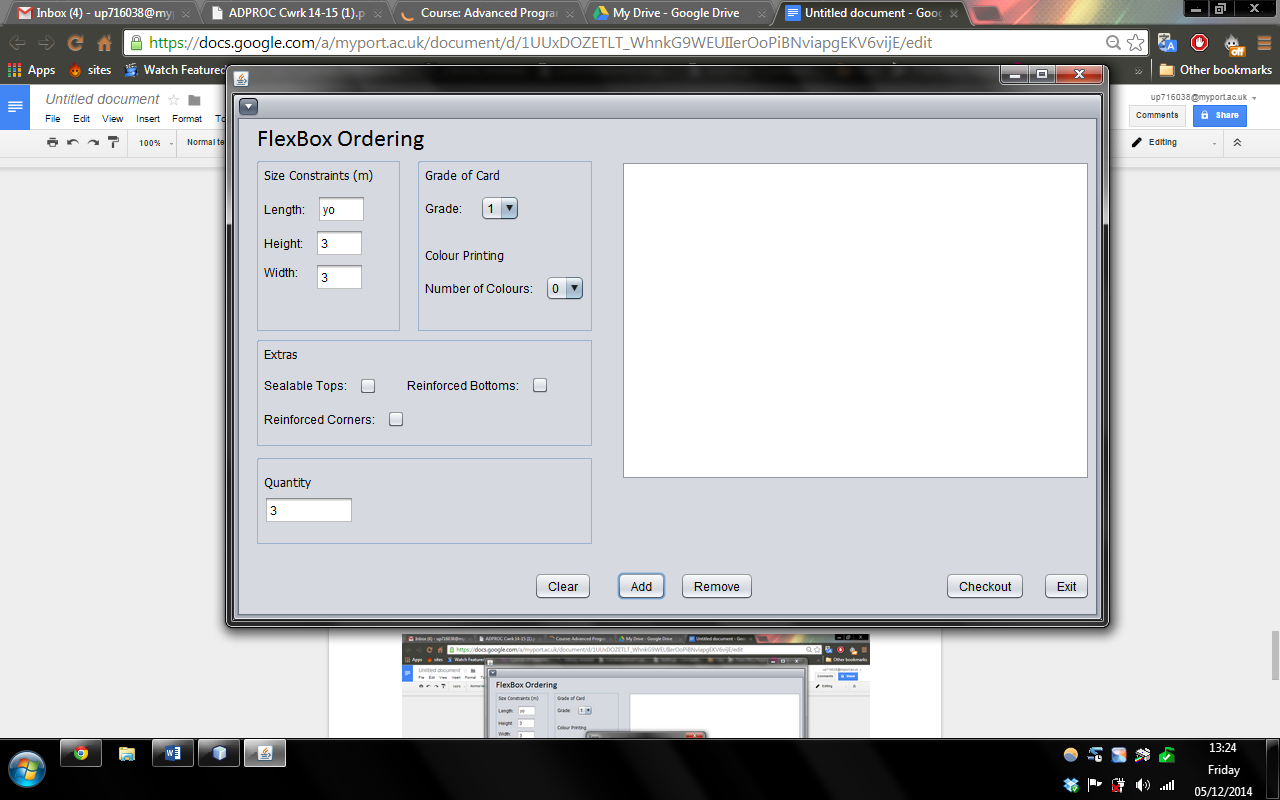


The length, height and width text fields all contain a value of 3.

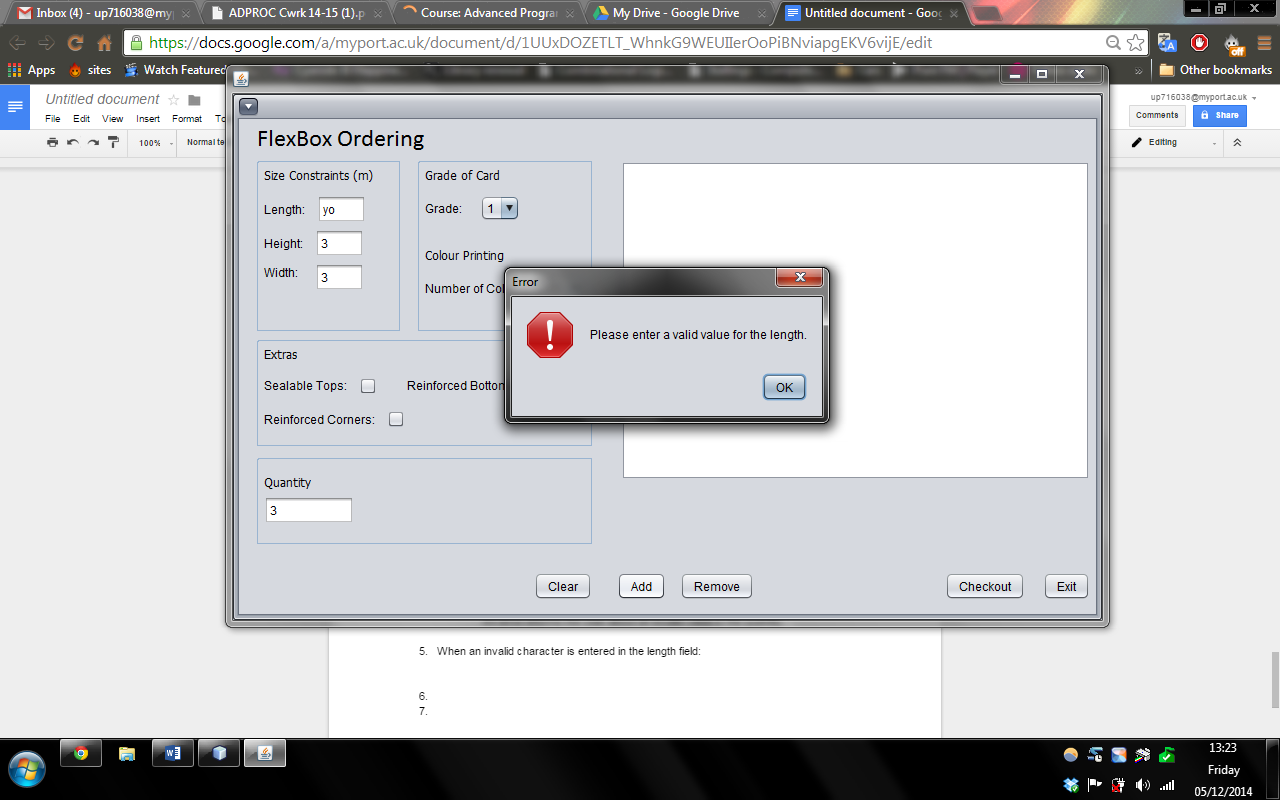


An error informs the user about an invalid value in the quantity.

1. When an invalid character is entered in the length field:



The length text field contains the value ‘yo’.

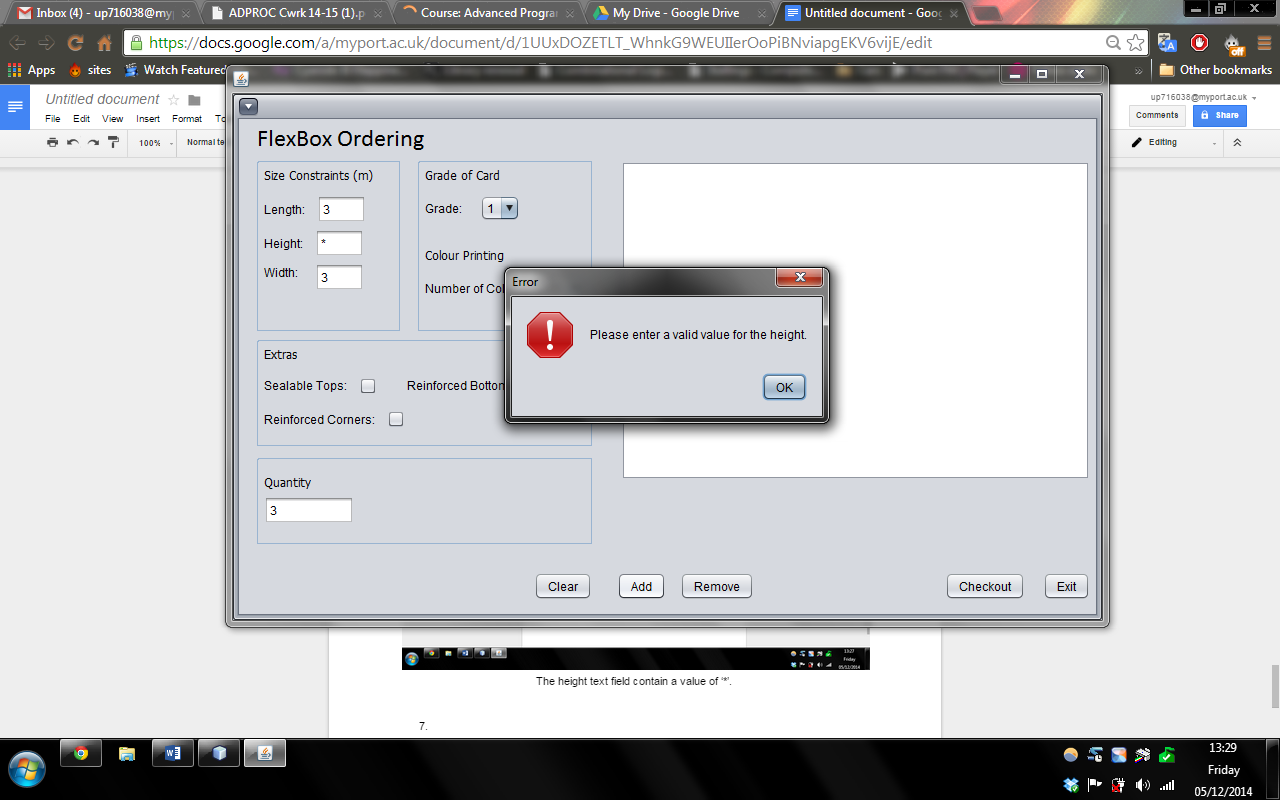


An error message informs the user of an invalid value in the length text field.

1. When an invalid value is entered into the height text field:

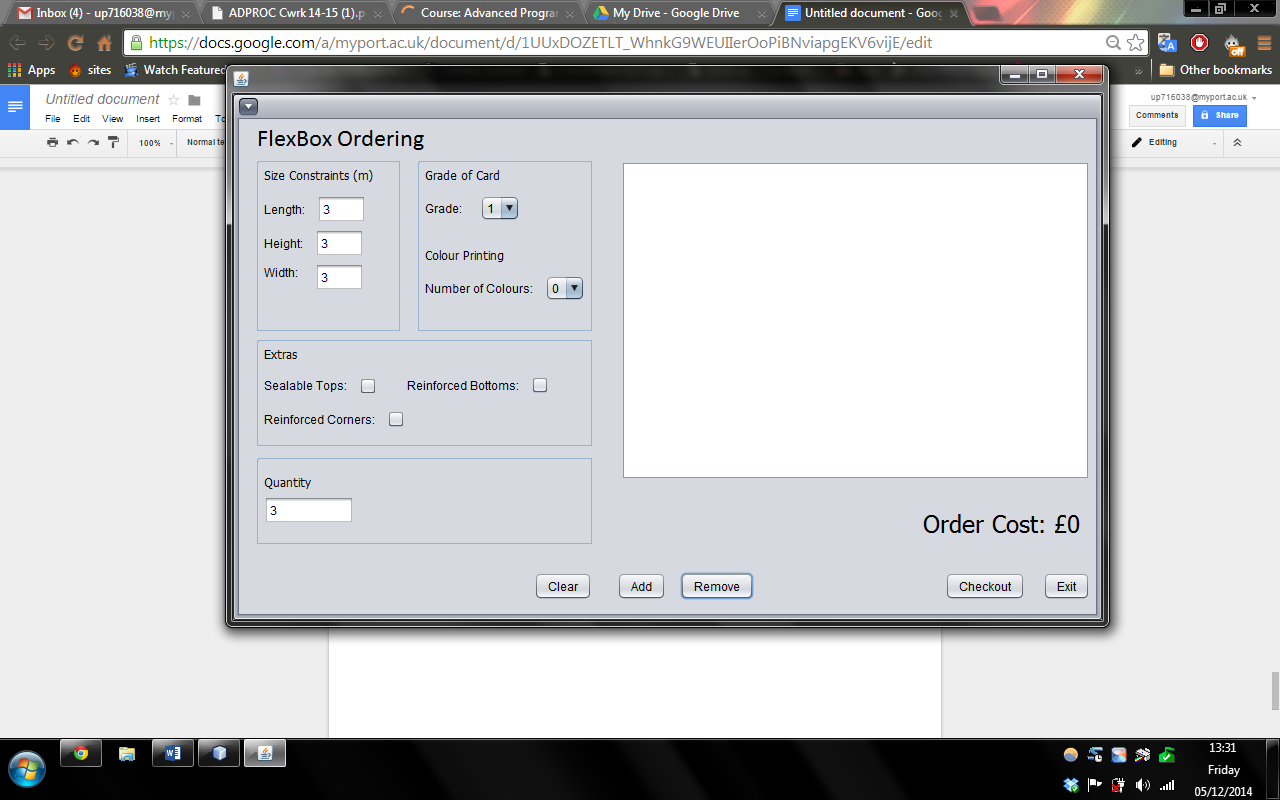


The height text field contain a value of ‘\*’.

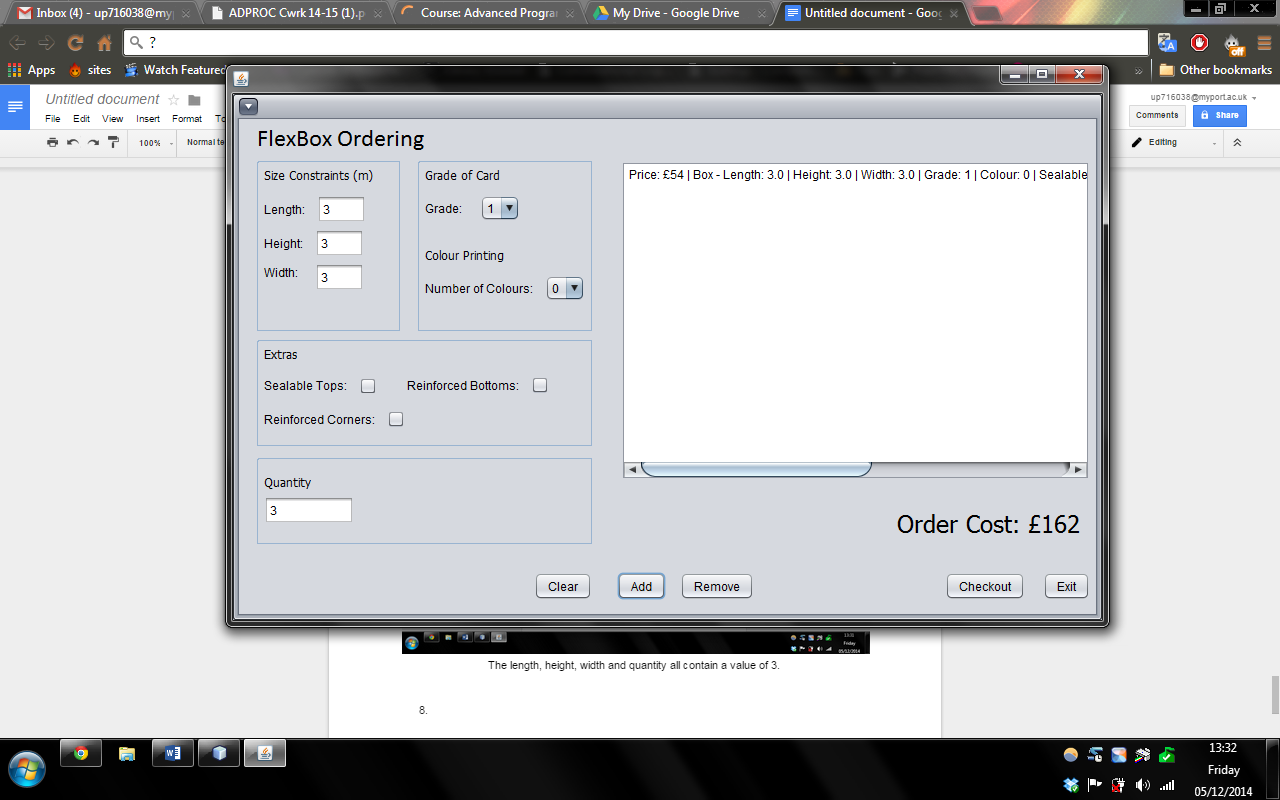


An error message informs the user of an invalid value in the height text field.

1. When all the parameters contain valid values:

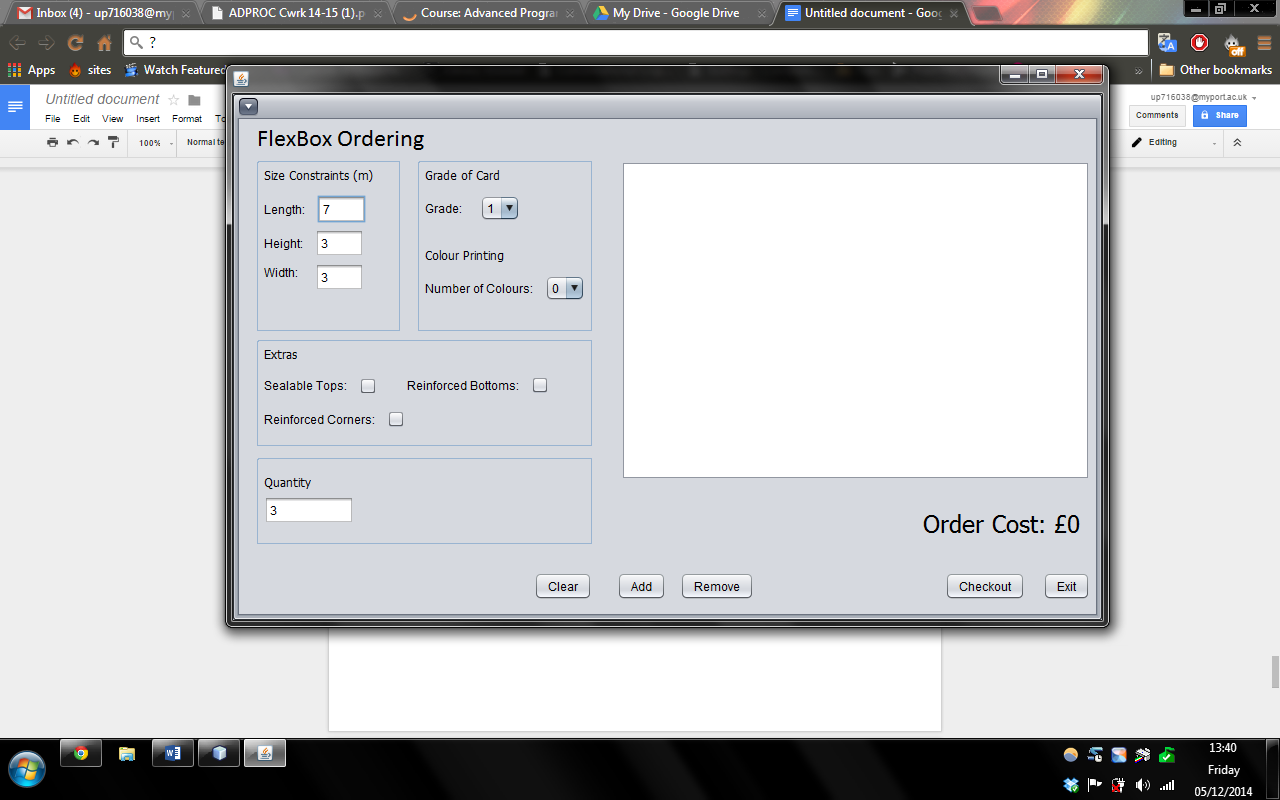


The length, height, width and quantity all contain a value of 3.

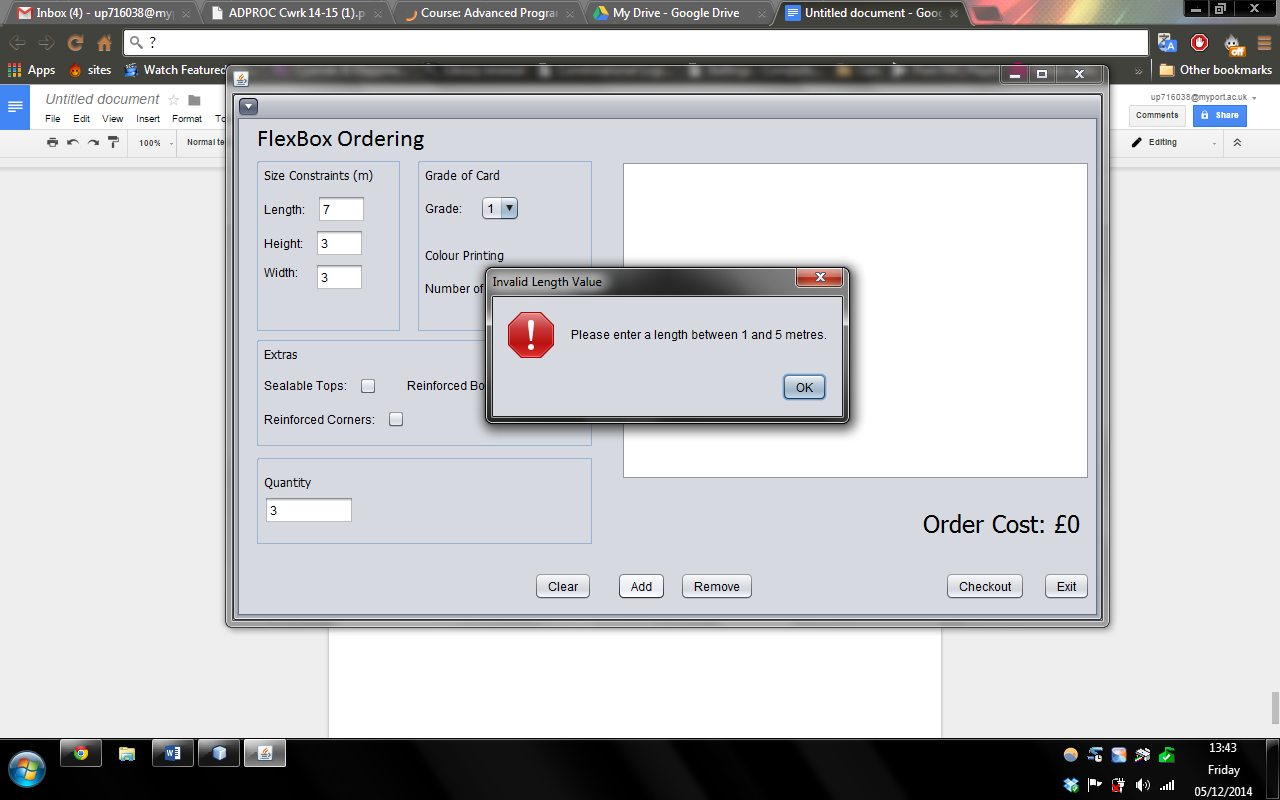


An order is added to the JList, showing all the attributes and the price of the order (quantity included in price).

1. When a value out of range is entered into the length text field:

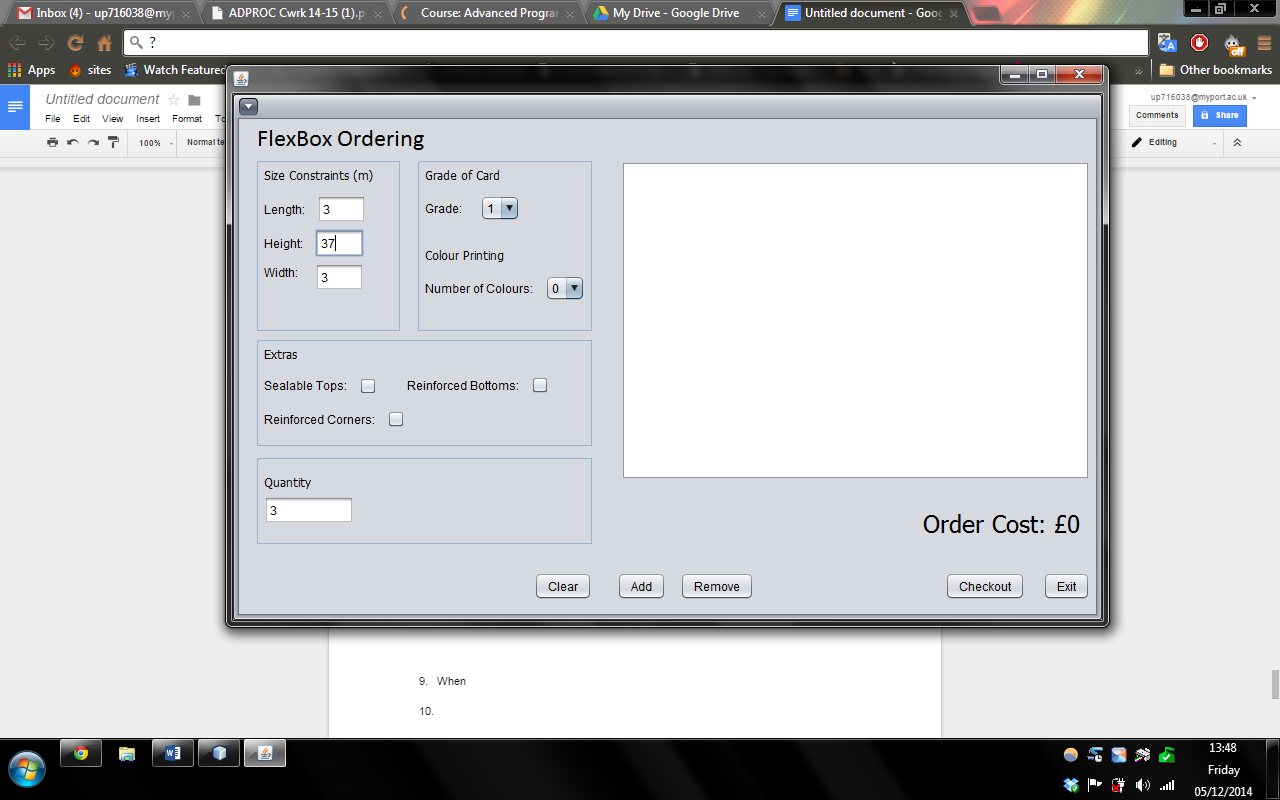


The length text field contains the value 7.

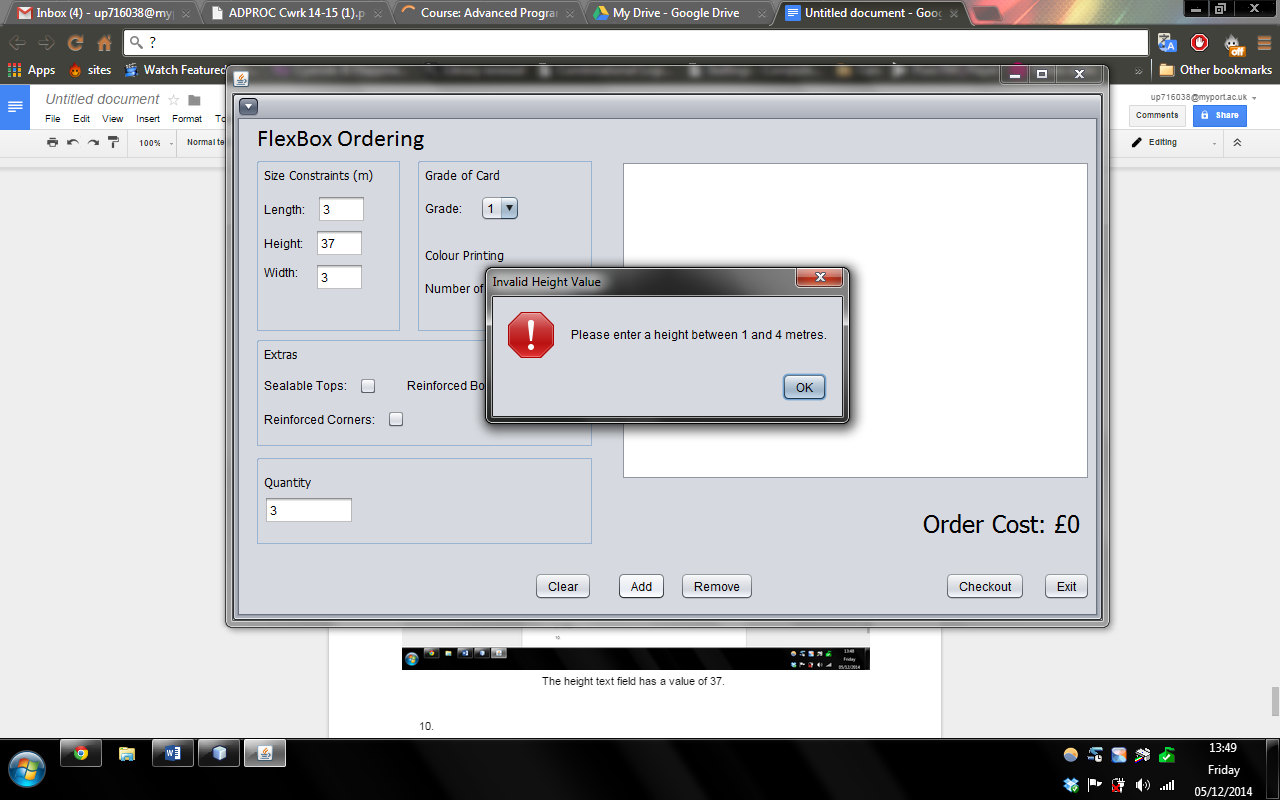


An error message pops up informing the user to enter a value within the range of the length.

1. When a value out of bounds is entered in the height text field:

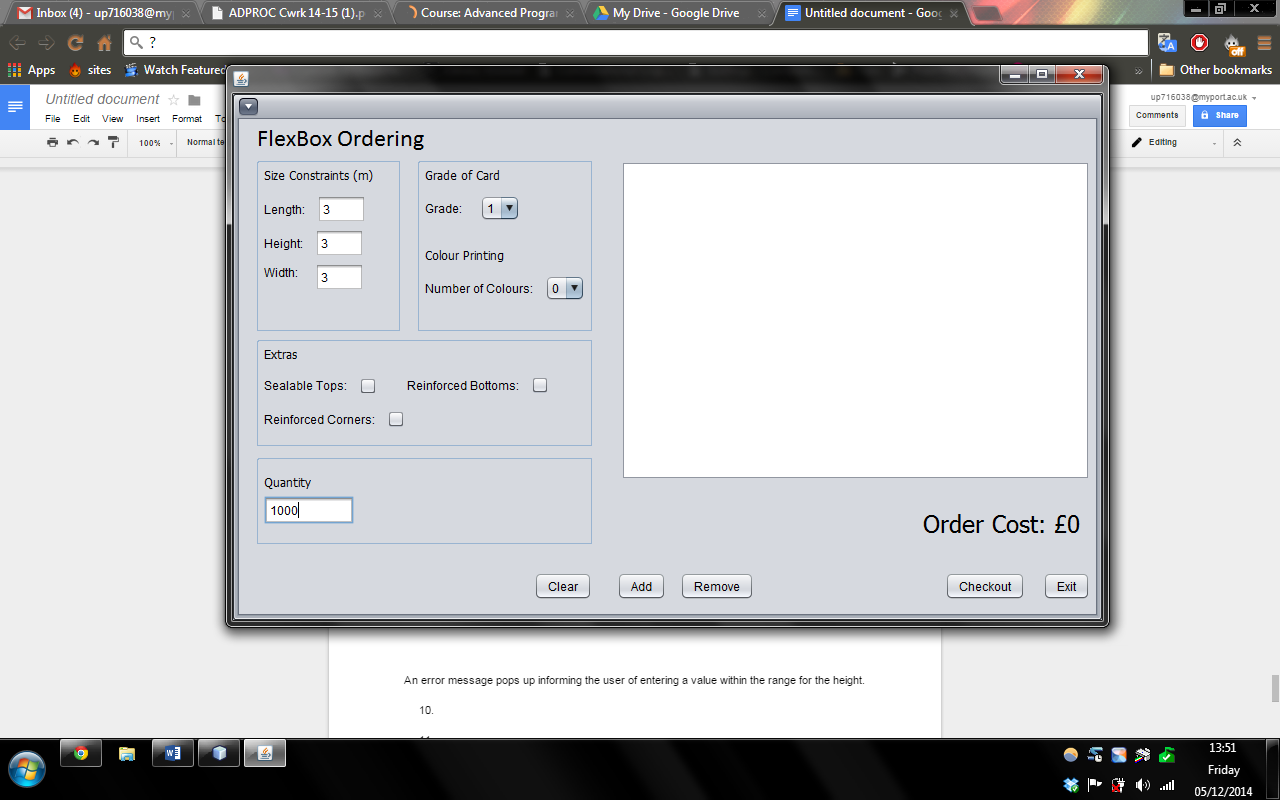


The height text field has a value of 37.

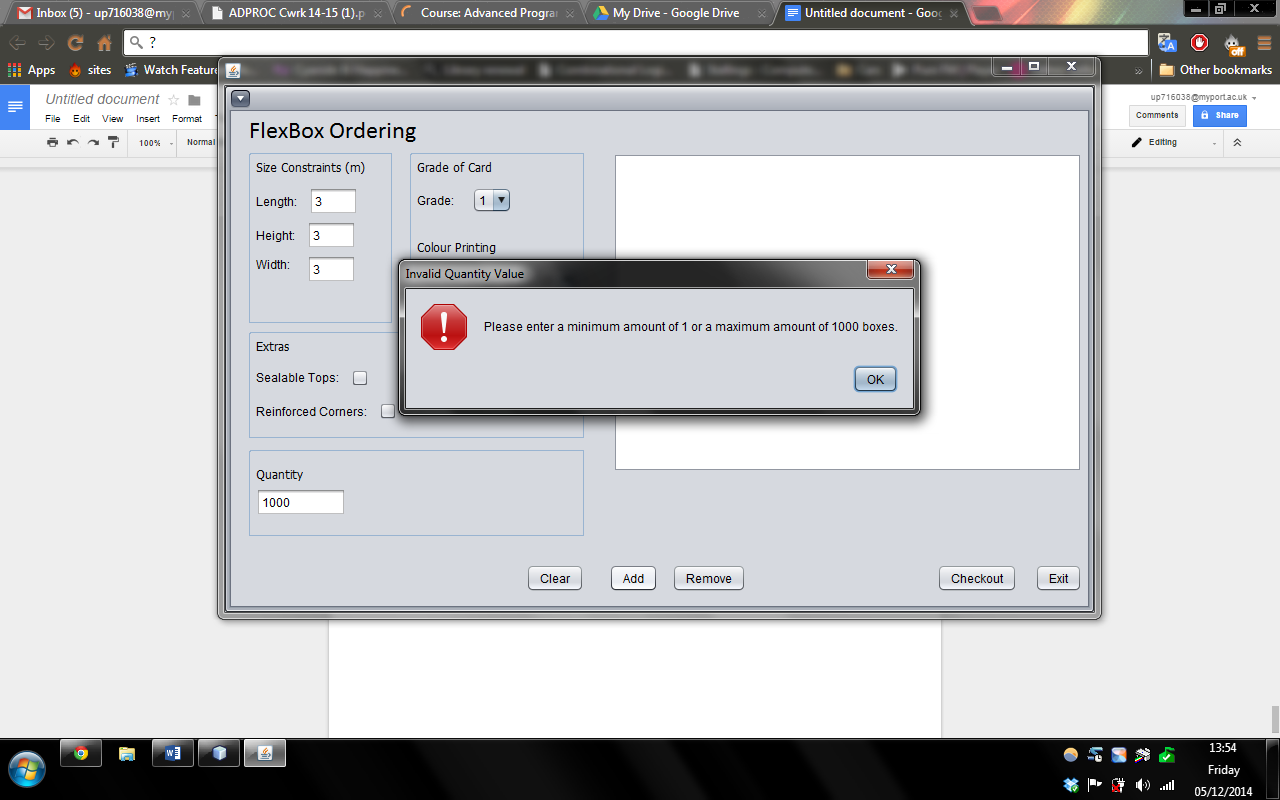


An error message pops up informing the user of entering a value within the range for the height.

1. When an out of bounds value is entered for the quantity:

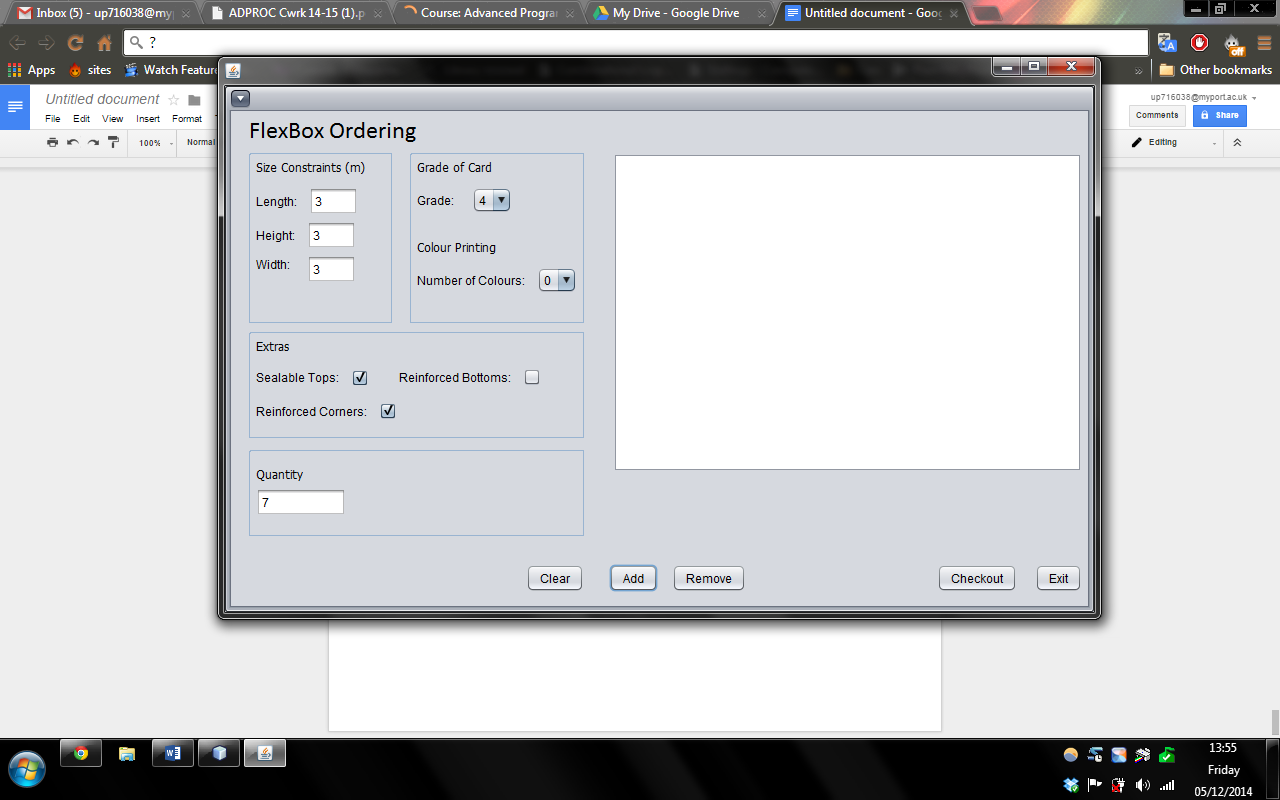


The value of the quantity text field is entered as 1000.

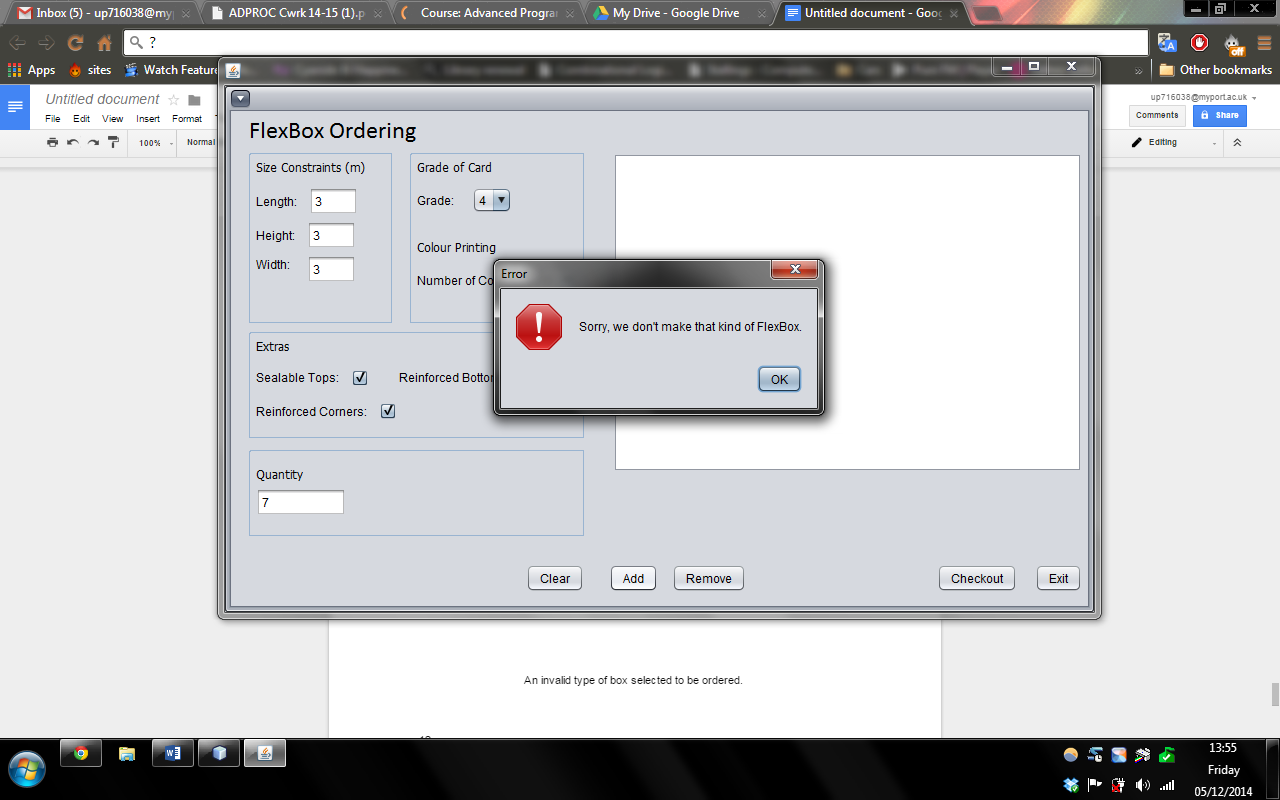


An error message pops up informing the user to enter a value within range for the quantity.

1. When an invalid type of box is ordered:

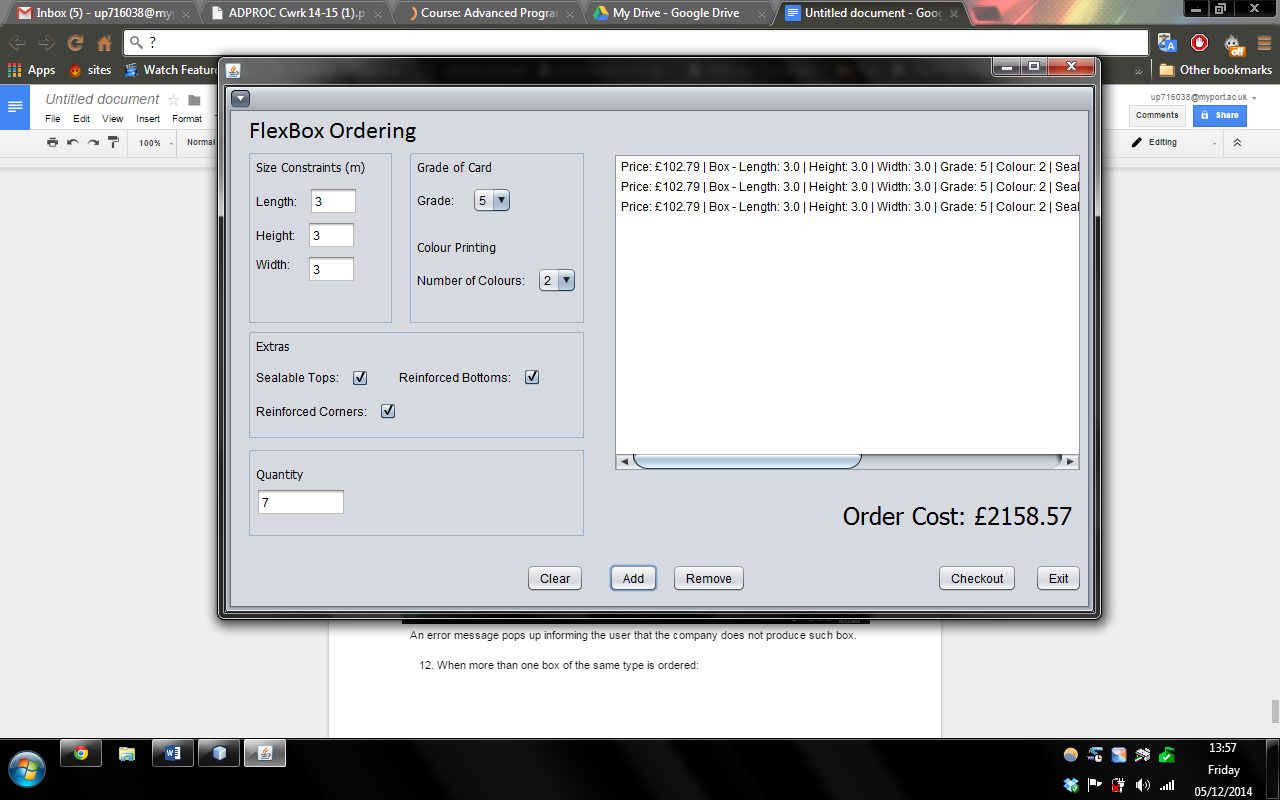


An invalid type of box selected to be ordered.



An error message pops up informing the user that the company does not produce such box.

1. When more than one box of the same type is ordered:



Everytime add is pressed, a single box order is added to the JList and displayed on the right, showing the total of the whole order at the bottom right.