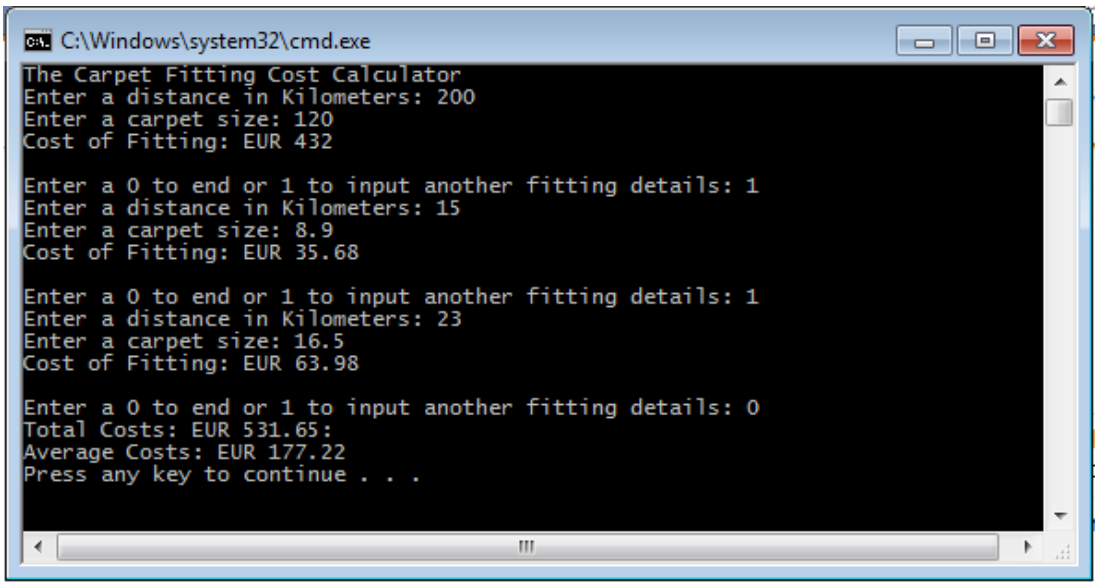
Date: Nov 7th 2017

IMPORTANT: All code should be in **one** solution file and uploaded to Moodle.

Question 1: Calculations/Methods/Loops (33%)

a) Write a program which calculates the total cost of fitting a carpet based on the carpet size, distance to travel to the fitting and fitting cost per square meter. The rates are as follows.
- Travel costs: €0.45 per kilometre
- Fitting cost per square meter: €3.25
- Both of these are constant values and your code should reflect this



b) Include 2 variables to hold the total cost of all fitting charges and the number of fittings.

c) In the Main() have the inputting of data repeat until a sentinel value is entered so that a number of delivery charges can be calculated.

d) Introduce a method to the program so that the fitting charge is calculated in a method and the cost is returned. If the fitting cost is over €250 then this method will apply a 10% discount. This method also updates the 2 variables in part (b).

e) After the sentinel is entered the total of all fitting charges is outputted as well as the average cost of each fitting. All of this should be done in a specific method with the required details passed to the method.

Marking Scheme

Comments/Readability/Output	3	
Appropriate types	3	
Part A	4	
Part B	3	
Part C	7	
Part D	7	
Part E	6	
Total		33

Date: Nov 7th 2017

Question 2: Classes (50 marks/67%)

a) Create a class called MemberTest and also a class called Member. In the Member class add auto implemented properties for the First Name, Last Name, Age, MemberID and MembershipStatus (that is either current or expired). Add a Constructor to take 4 parameters, MembershipStatus is not a parameter but is set to "expired" within the constructor. Override ToString() to display all the Member Information. In the Main() in Member Test create 5 Member objects using the Constructor and add the Member objects to an array of Members.

b) In MemberTest create a method that takes an array of Member objects as a parameter. This method will, using a loop, output all the details for all of the members. The Main() will then request this method to display the details for all the members.

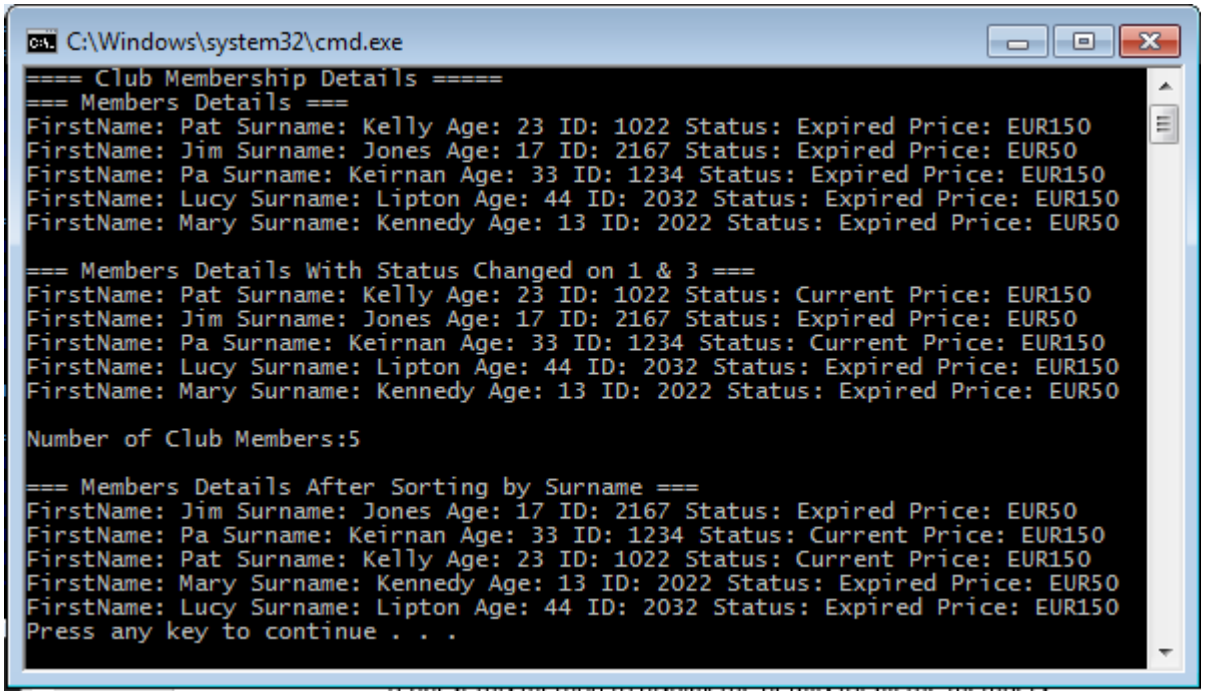
c) In Member add a method called ChangeStatus which inverts the status when called. Have the main program change the status of the 1st and 3rd member objects. Output all members details again using the method in part (b).

d) Add a method to set a price field for a member, this price field has a read only property Price. The call to set the Price is made by the Constructor. Member price is different depending on age. If a member is younger than 18 the price is €50. If older than 18 the price is €150. Update the ToString() method in Member to take account of this new property.

e) Add a static variable which stores a count of Members, this variable is updated in the Constructor. Output this value from Main().

f) Implement IComparable and code the CompareTo method so that Members objects can be sorted by Last Name. The Main() method should request that the Array of Members is sorted in reverse order. Display the members' details after sorting.

Date: Nov 7th 2017



Marking Scheme

Comments/Readability/Output	3	
Appropriate types	2	
Part A	14	
Part B	8	
Part C	10	
Part D	12	
Part E	5	
Part F	13	
Total		67