***Group No: ?***

***Introduction***

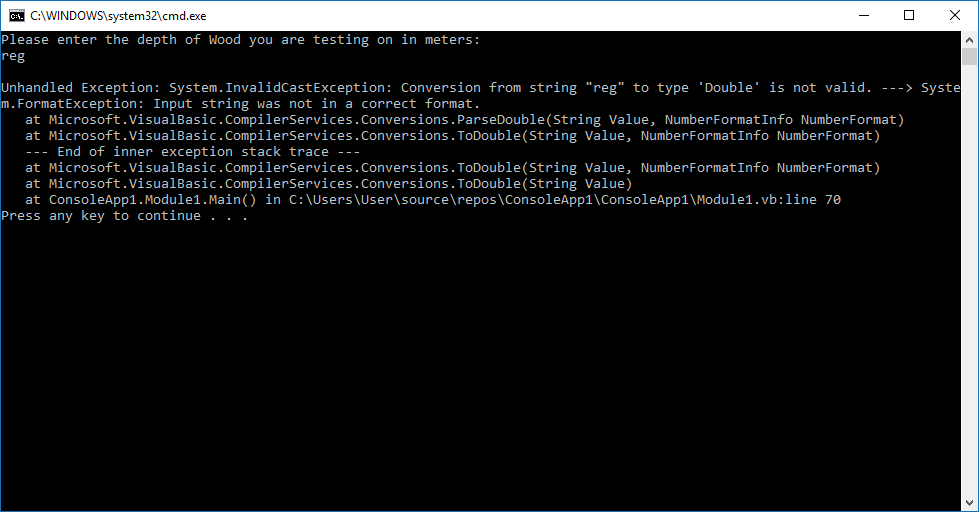
This program is designed to be helpful to the scientists who will be using it, asking them to input depths, checking whether the are valid field results, aka above 0; and then going throw a secondary verification process of confirming with them the value they input. the program will then inform the scientists the period the bullet spent in the medium, and then the average times over the 3 materials the bullet spent in the medium.

***Question 1***

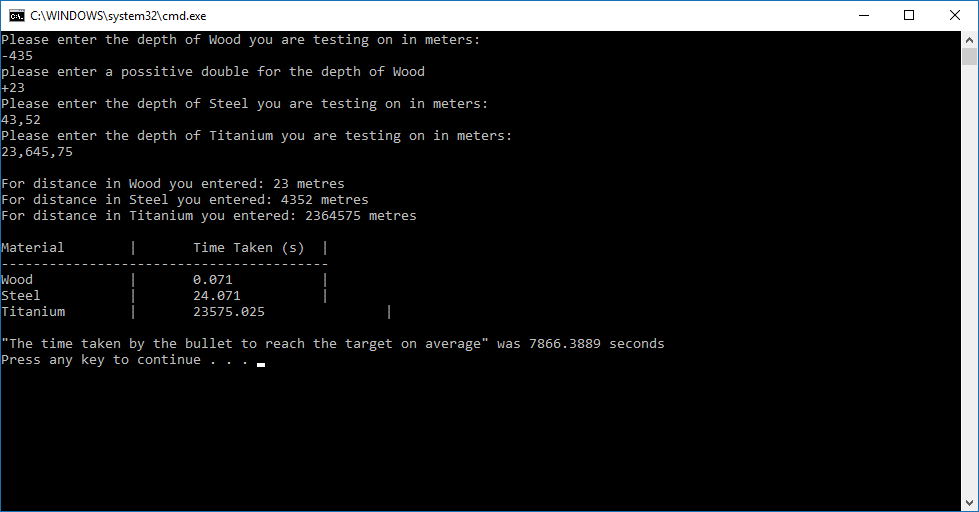
*Introduction: functionality of the program*

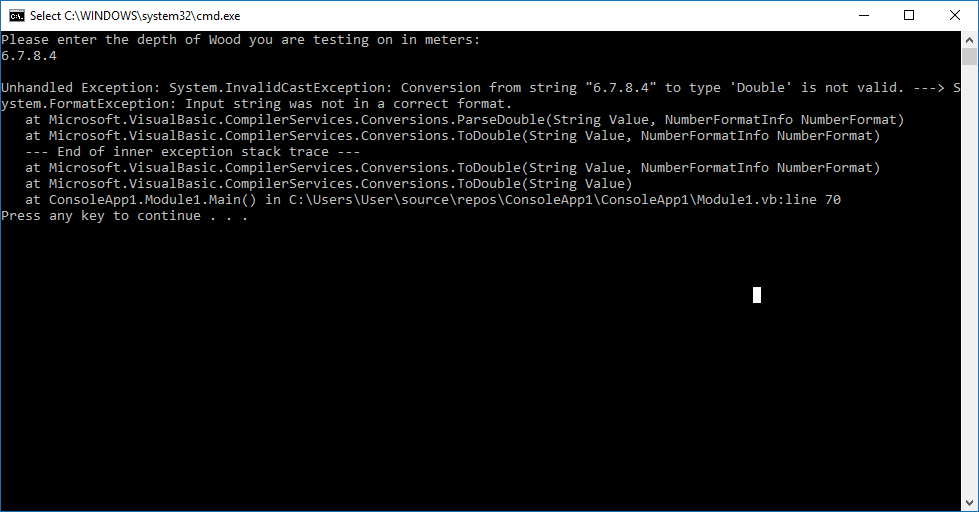
1. *any special instructions or warnings to the user (or assessor!) such as in which case code does not work ,*

* Any value other than a numerical, point or comer and it will output an “Unhandled Exception:”, complaining about not being able to convert the inputted string to type double. This means the end user should not input any units after the numerical data; having converted to standard SI units (metres) beforehand. (evidence in bellow screenshot)



* The only other Characters other than integers which are accepted are comer which but is removed during this string to double type conversion, thusly the calculated for time will take the depth as inputed without the commer (evidence in bellow screenshot). Fallstopes (decimal places) and “-” & “+” can also be inputed however their can be no more than one fallstop in the users input as this will crash because it carnt convert a string with multiple fallstopes into a double (evidence in the second bellow screenshot)





* The user must input a depth greater than 0, other wise they will be asked to input another value which is above 0/a positive double (evidence in above screen shot)

*e.g.* ***Any user input validation***

***e.g. Any limitation of the program***

*e.g.* ***Evidence: Tested by the user input …. and the result is ….***

1. *or to draw attention to any aspects of the program that you are particularly proud of (i.e. reusability, maintainability aspects of the code.)*

I’m particularly proud of my nonspecific up to 5 coulomb header generator isolates the heading formatting to one place rather than having to type it our every time. It also seems intuitive to use, how many coulombs do you want and then a list of what you want to put inside the coulomb.

I’ve also written functions for every equation which should aid those who don’t intuitively know that t = s/v for instance. Allowing them to manipulate the code in the future which affecting the original equation by accident.

***e.g. use of variables and why , what functions are used and why..***

*Conclusion:*

In conclusion I have implemented several functions to aid future programmers re-write this code as well as writing new programmes, taking cognitive load off their minds whilst coding because possibly long complicated calculations have been graphically simplified to a function name. My functions have also been useful to those writing q2 to q4. The functions them selves were written to be as general as possible, thusly being useful in more programs, and being written simply so that they don’t do to much and become more complicated for the user to understand as keep them generally usfull

*e.g. all/ essential/ core/ a few required functions of the question 1 are successfully implemented.*

*use of functions /are challenging /makes it easy /…..*

***Question 2:***

*---------------------*

***Question 3:***

*---------------------------*

***Question 4:***

*----------------------------*

***Overall Conclusion***

*e.g.*

*all/ essential/ core/ a few/ required functions of CW are successfully implemented. There is some initial problems with ….. However later it has been …….*

*use of functions /makes it easy to develop/ cause many problems*

***Word count***