Roots Programming Assignment
Numerical Analysis

Goal: Create a project which can use either the Bisection Method or Newtons Method to approximate the root of a function.

Steps to follow - in this order:

1. Open the programming project:
   - Navigate to https://github.com/pattiscot/Numerical-Analysis-Student-Ready.git
   - Click the green "code" button and select "Download file"
   - Unzip the folder and open the RootsApp. Simply double clicking the .zip file to open it will NOT suffice. It will not work properly in Visual Studio if you do this.
   - Double click the RootsApp.sln file. This should open the project up in Visual Studio. You will need to use version dotnet 8.0 for all your programs this semester.
2. Identify and inspect the interface.
   - You will find the Solution Explorer on the right hand side bar of Visual Studio. Use the Solution Explorer to find the interface which you need to implement. Recall that all interfaces are named with a leading capital I. So the interface in this project is a file named IRoots.cs. You will note a similar interface which I implemented name IRoot.cs. This should be used by your program to return various messages when things do not work right in your program. You should NOT make any changes to any of the interface classes (or other classes except the Programs.cs class and the class you create to implement the interface). However, you can inspect the interface to see what you are supposed to do in your implementation. You can open the interface (or any file in the file explorer) by double clicking the name of the file in the Solution Explorer.
3. Add the implementation class to the project
   - Right click the name of the project (the bold name in the Solution Explorer) and select ADD and then in the sub-menu, select Class...
   - Select the class option and then, in the bottom area of this window pane, name the class Roots.cs
   - Click Add
4. Setup the inheritance **Finish from here!**
   (a) If the implementation class you just created does not automatically open then double click its name in the Solution Explorer to open it.
   (b) Replace the word internal with the word public. This changes the class to a public class which has greater accessibility within the project.
   (c) Setup the inheritance from IRoots.cs by adding :IRoots to the end of the line "public class Roots" like you learned to do in one of the tutorials.
   (d) Click the little light bulb - usually on the left hand side of the screen - and tell it to implement the interface. This should copy all the functions that I specified in the interface to your implementation and give you space to write your code.
5. Determine what you need to program by reading the interface.
   - The BisectionMedthod method implements the Bisection Method to find the zero of a specified function in a specified interval. This method returns an IRoot object and requires the user to provide the following
     – double a - the left hand endpoint of the specified interval
     – double b - the right hand endpoint of the specified interval
     – Func¡double, double¿ f - the specified function which takes a double and returns a double. This is the function that you want to find the root of in the specified interval.
     – double epsilon - a very small number which you will use to determine when to break out of the loop. According to the Bisection Method, when your new left- and right-hand endpoints are close enough, then you have the approximate zero. This value specifies how close the endpoints need to be.

  - int nmax - the maximum number of times you want to iterate through the loop before you break it and return one of our error values from Root.cs
- The NewtonsMethod method implements Newton's Method to find the zero of a specified function near a specified initial x value. This method returns an IRoot object and requires the user to provide the following
  - double a - the specified initial x value
  - Func¡double, double¿ f - the specified function which takes a double and returns a double. This is the function that you want to find the root of near a.
  - Func¡double,double¿fPrime - the derivative of f. This is used in the calculation of Newton's Method.
  - double epsilon - the accuracy desired. You stop when the difference of consecutive approximations to the zero is less than epsilon.
  - int nmax - the maximum number of times you want to iterate through the loop before you break it and return one of our error values from Root.cs

6. You can use the Program.cs file for all your I/O. This can be used to help debug your program.
7. Submit your program
   - Create a zip file of your Roots.cs file.
   - Using the link to OICLearning posted on eclass, navigate to the Roots App assignment.
   - Submit this zip file on OICLearning. Wait a little while (up to a few minutes) to see your grade. You can resubmit as many times as you want until the deadline is passed. I will record your last score on eclass - so make sure that this is your highest score.