Exercise 14-1: The Median

STEPS:

Populate int array[] and determine array’s const int SIZE for passing

Create function called median that accepts array and size as parameters

Ex. median(int array[], const int SIZE)

\*review notes before continuing\*

Verify that the array is not empty

Ex. (SIZE > 0)

If array is empty then throw numeric exception value of -1

Verify if elements in array are sorted from lowest to highest

If array is not properly sorted then throw numeric exception value of -2

Otherwise, calculate and display the median value in the array

Demonstrate median function in main and catch any thrown errors

Error of -1 should display message: “Error: No elements in array.”

Error of -2 should display message: “Error: Array not in sorted order.”

\*Notes\* The median of an array of odd numbers is the integer in the middle of the array, but the median of an array with even numbers is found by taking the two numbers in the middle and diving by 2 to determine the average of the numbers.

\*Hints\* Use the “Using Arrays Outside Main Function” for reference.

Exercise 14-2: Common User Error

Create function called getnum(\*int x\*) that accepts a value (x) to verify it is numeric

If the value (x) is numeric it is returned

If value (x) is not numeric the program should \*recover from the error status\* then throw an exception

In main function create a Try…Catch which uses the getnum() function to get an integer

If the user inputs character data, getnum() should display an error message

Program should loop until a numeric response is given

\*Hints\* If user inputs char instead of int, the statement should be false. Easy check for numeric data could be if (cin >> num) to determine if input is invalid. \*Also there is potential to use while(!(cin >> num)).\*

\*Hints\* Research cin.clear() and cin.ignore() statements. If input results in error, the input status must be reset and the buffer must be cleared out before asking for input again.

\*Hints\* Use the YouTube video “Input Validation C++ (integer) – while() – cin.ingore() – cin.clear()” for help. This video is almost exactly what I have to do.