Programming Exercise 03
Errors and Exceptions

ISTA-220, C# Step by Step

This activity consists of four programming exercises. The following exercises are open book and open note. You are free to use any written documentation you wish. However, these are individual exercises, and you cannot consult with each other in writing your programs.

This programming exercise has four parts consisting of four requirements. The grade for each requirement is indicated, for a maximum of 100 points. At a minimum, your program must compile successfully and run.

This exercise builds on Exercise 1, Mathematical Formulas. You may have noticed that some inputs do not work, and that sometimes users make errors in inputs. In this exercise, we will address some of these issues.

Circles, hemispheres, and triangles: 70 points For these formulas to work, you must enter a number consisting of digits — alphabetical characters will not work. For these three parts, add a try/catch block for each formula that handles a FormatException error. Additionally, for these formulas to return valid values, input must be greater than zero. Add a try/catch block for a general exception using a filter or some other technique to guard aghainst users entering numbers equal to or less than zero.

Comprehensive exceptions, quadratic formula error: 80 points Modify your code so that your exception handling logic applies to inputs generally for all formulas, rather than in dividual try/catch blocks for each formula. In addition, the quadratic formula as an obvious problem: you can't take the square root of a negative number ... at least until we implement imaginary numbers. Modify your program to prevent taking the square root of a negative number.

Variable overflow: 90 points It's possible that some values may overflow the numeric conctraints. Modify your code (by writing checked/unchecked statements) to protect against numeric overflow errors.

Finally block: 100 points Modify your program to add a finally block. This actually doesn't have to do anything, as we haven't covered network connections, file handles, etc. Just have it print a message such as, "This program has finally terminated."