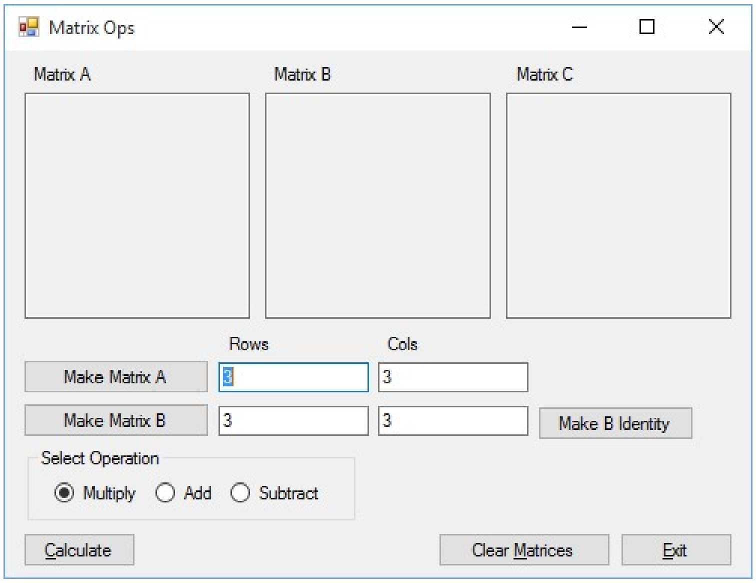
CIS 3309 Section 2 Lab: Matrix Ops Problem

The purpose of this lab is to test your knowledge of C# up to Section 2, Chapters 1 to 11. In this exercise, you will code an application that makes two random matrices and performs multiplication, addition and subtraction operations. Please begin by creating a project called “matrixOps” and creating the form as shown below. Make sure to give form components, data and methods meaningful names, and use comments for the program description, methods and significant blocks of code. You will also need to declare 3 matrix identifiers as class data to contain the A, B and C matrices.



Make Matrix A: Rows 3, Cols 3
Make Matrix B: Rows 3, Cols 3
Select Operation: ● Multiply ○ Add ○ Subtract

The boxes labeled Matrix A, B and C are read-only multiline textboxes with size property set to 150 by 150. The textboxes for buttons Make Matrix for A and B must be validated as an Int32 during the Leave event (Hint: try to convert the text property into an Int32 and use Try-Catch). The user should not be able to leave the TextBox until a valid integer is entered. The initial value of the Text property for all Rows and Cols should be set to 3 and the cursor should be placed in the Matrix A Rows Textbox when the form loads. Set a reasonable tab order to start at the Rows TextBox for Matrix A and tab through the components to allow efficient entry of data and execution of operations for the application.

The Make Matrix button click events for A and B should call a method to generate random matrices of dimensions recorded in the Rows and Cols textboxes. For example, if you click Make Matrix A, given the state of the form above,

1. A method should be called to create a 3 x 3 A matrix with random doubles
2. Another method should be called to convert the matrix to a string and copy it to the textbox

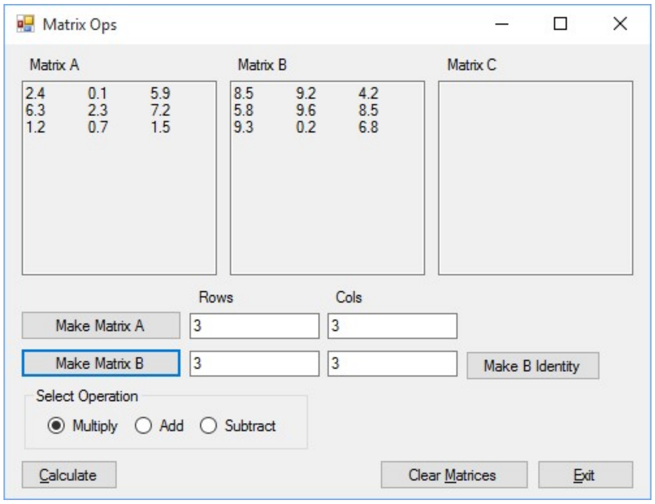
This should be the same for Make Matrix B. Make a single method to create the matrix for both A and B and another method to copy the matrices to the Textboxes. Try to eliminate redundant code. To get a random double declare a random object:

```
// Declare a random
Random autoRand = new Random();
```

And use it to get a random and put it in the matrix.

```
// Get next random and put in matrix
matrix[i, j] = autoRand.NextDouble() * 10.0;
```

To copy a matrix to a TextBox, you need to pass the matrix and TextBox as arguments. After making both matrices, the form should appear as below (the doubles in the matrices should have different values each time a matrix is created).



Matrix A:

```
2.4    0.1    5.9
6.3    2.3    7.2
1.2    0.7    1.5
```

Matrix B:

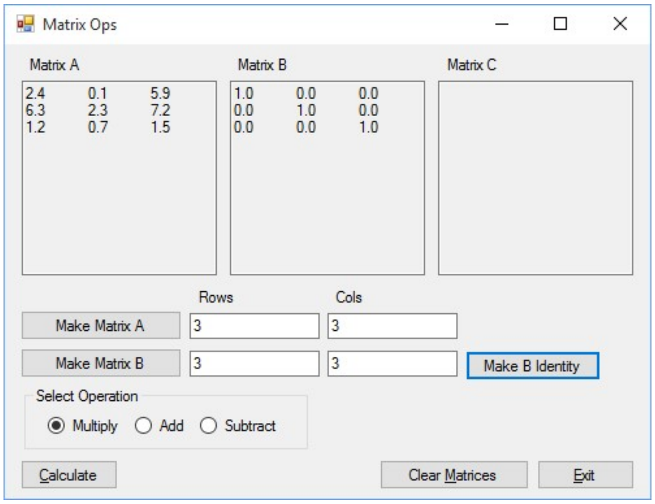
```
8.5    9.2    4.2
5.8    9.6    8.5
9.3    0.2    6.8
```

Make Matrix A — Rows: 3 Cols: 3

Make Matrix B — Rows: 3 Cols: 3

● Multiply ○ Add ○ Subtract

The Make B Identity should convert the B matrix to the identity matrix only if the number of rows and columns are equal and copy it to the Matrix B TextBox. See the form below. Note all but the diagonal elements in B are zero and the diagonal elements are 1.



Matrix A:

```
2.4    0.1    5.9
6.3    2.3    7.2
1.2    0.7    1.5
```

Matrix B:

```
1.0    0.0    0.0
0.0    1.0    0.0
0.0    0.0    1.0
```

Make Matrix A — Rows: 3 Cols: 3

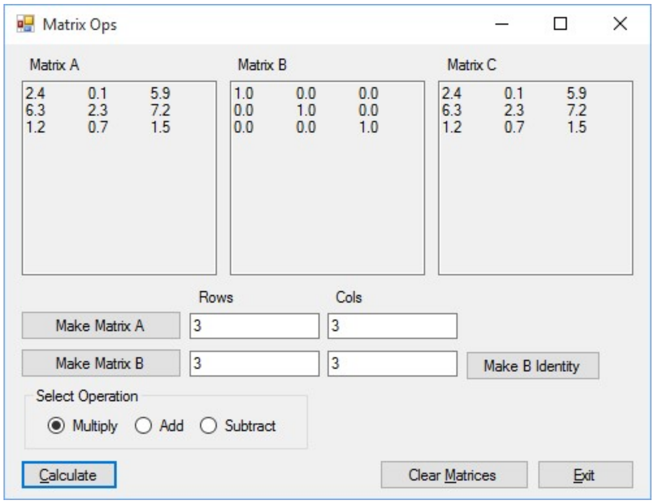
Make Matrix B — Rows: 3 Cols: 3

● Multiply ○ Add ○ Subtract

The Select Operation GroupBox has 3 RadioButton objects to multiply matrices, add matrices and subtract matrices, respectively. The Multiply RadioButton should be checked when the form loads (Hint: double-click on the form—What event is created?). Make sure that only one RadioButton can be selected at a time. The calculate Button should perform the operation selected in the Select Operation GroupBox. Make sure that no operation can crash your application and that all operations can only execute correctly. You can use a validation approach or handle the exceptions. In any case, provide a clear message to the user what went wrong and how to correct the problem. Things to consider:

1. The A and B matrix are not null
2. For multiplication, the rows of A match the columns of B and the columns of A match the rows of B
3. For addition and subtraction the A and B matrices should have the same dimensions.

When calculate is clicked, the C matrix should be created with the proper dimensions to contain the result of the operation given the sizes of A and B. After selecting Multiply and clicking Calculate, the form should have the result in the Matrix C TextBox as shown below.



Matrix A

```
2.4     0.1     5.9
6.3     2.3     7.2
1.2     0.7     1.5
```

Matrix B

```
1.0     0.0     0.0
0.0     1.0     0.0
0.0     0.0     1.0
```

Matrix C

```
2.4     0.1     5.9
6.3     2.3     7.2
1.2     0.7     1.5
```

	Rows	Cols
Make Matrix A	3	3
Make Matrix B	3	3

Select Operation

● Multiply ○ Add ○ Subtract

Please run the sample app included with this assignment to get an idea of how the program should work. When done coding, please test your app, close the solution, zip it and upload it to Blackboard.