Name: **Session:**

Programming II

Lab Exercise 4.4.2023

In this application, you will need to use a Windows form. You may use my design or create your own.

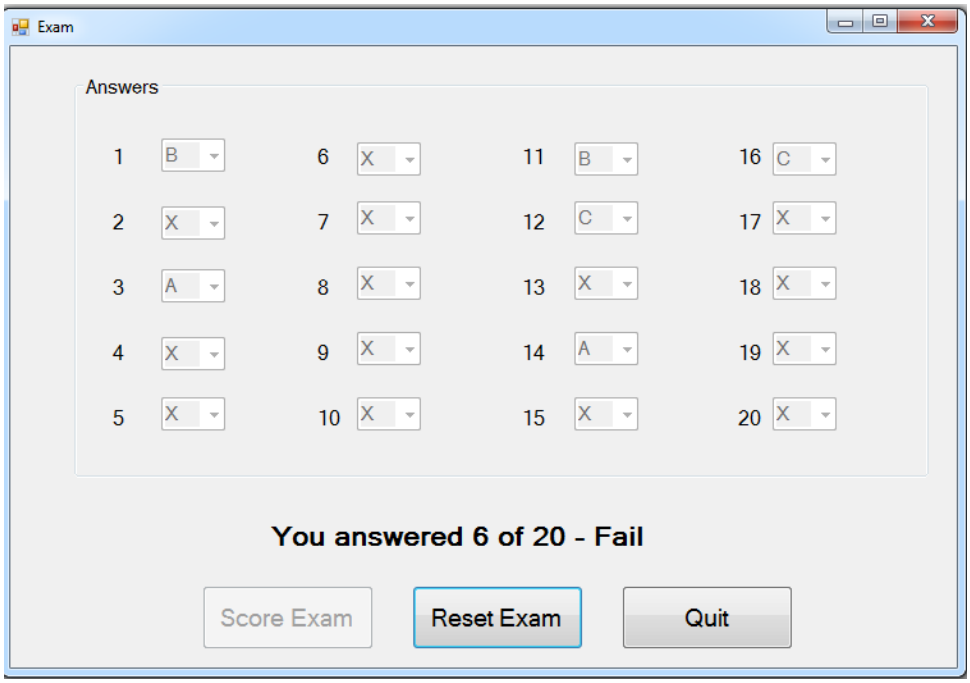
1. The local Registry of Motor Vehicles office has asked you to create an application that grades the written portion of the driver's license exam. The exam has 20 multiple choice questions. Here are the correct answers to the questions.

   1. B 2. D 3. A 4. A 5. C 6. A 7. B 8. A 9. C 10. D
   11. B 12. C 13. D 14. A 15. D 16. C 17. C 18. B 19. D 20. A

   Your application should store the correct scores in an array. A form should allow the user to enter answers for each question.

   When the user clicks the Score Exam button, the application should display whether each question was answered correctly or incorrectly and whether the student passed of failed the exam. A student must correctly answer 15 of the 20 questions to pass the exam. Only accept letters A, B, C, and D as answers. If you try to score your exam without answering all questions, the questions you answered will be scored and you will be given the opportunity to answer questions until you have provide 20 valid answers.

   The application should have a reset and quit button.



a. Add the following code to the btnReset_Click event handler.

```
ComboBox[] scores = new ComboBox[] {ComboBox1, ComboBox2, ComboBox3,
    ComboBox4, ComboBox5, ComboBox6, ComboBox7, ComboBox8, ComboBox9,
        ComboBox10, ComboBox11, ComboBox12, ComboBox13, ComboBox14,
ComboBox15, ComboBox16, ComboBox17, ComboBox18, ComboBox19, ComboBox20};

for (int index = 0; index <= 19; index++)
{
        scores[index].Enabled = true;
        scores[index].Text = "";
        scores[index].ForeColor = Color.Black;
}

btnScore.Enabled = true;
lblResult.Text = "";
numCorrect = 0;
valid = 0;
```

b. Add the following code to the btnScore_Click event handler.

```
ComboBox[] scores = new ComboBox[]  {ComboBox1, ComboBox2, ComboBox3,
    ComboBox4, ComboBox5, ComboBox6, ComboBox7, ComboBox8, ComboBox9,
    ComboBox10, ComboBox11, ComboBox12, ComboBox13, ComboBox14,
ComboBox15, ComboBox16, ComboBox17, ComboBox18, ComboBox19, ComboBox20};

string[] correct = new string[] {"B", "D", "A", "A", "C", "A", "B", "A",
        "C", "D", "B", "C", "D", "A", "D", "C", "C", "B", "D", "A"};

for (int index = 0; index <= 19; index++)
{
        //check if answer is valid
        if (scores[index].Text != "")
        {
                valid++;
        }
        else
                continue;

        //Check if valid answer is correct
        if (scores[index].Text == correct[index])
        {
                numCorrect++;
                scores[index].Text = scores[index].Text;
                scores[index].Enabled = false;
        }
```

```
                //check if answer is wrong
                else
                {
                        scores[index].ForeColor = Color.Red;
                        scores[index].Text = "X";
                        scores[index].Enabled = false;
                }
        } //end of for loop

        //check for all questions answered
        if (valid == 20)
        {
                //check for pass
                if (numCorrect >= 15)
                   lblResult.Text = "You answered " + numCorrect.ToString() + " of 20 - Pass";
                //check for fail
                else
                   lblResult.Text = "You answered " + numCorrect + " of 20 - Fail";

                btnScore.Enabled = false;
        }
        else
        {
                lblResult.Text = "You have not answered all of the questions";
                valid = 0;
                numCorrect = 0;
        }
```

c. Test your program

In this lab you will create an application that uses a 2D array.

Theater Seating. Write a program that can be used to sell tickets for performances. The auditorium has 15 rows of seats with 30 seats in each row. The program should display a screen that shows which seats are available and which are taken. For example:

Seats

```
          1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Row 1     * * * * * * * * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 2     * * * * * * * * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 3     * * * * * * * * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 4     * * * * * * * * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 5     * * * * * * * * * *  *  *  * * *  * * * *  * * X * * * * * * * * * * * * * * *
Row 6     * * * * * * * * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 7     * * * * * * * * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 8     * * * * * * X * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 9     * * * * * * * * * *  *  * X * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 10    * * * * * * * * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 11    * * * * * * * * * *  *  *  * * *  * * * *  * X * * * * * * * * * * * * * * * *
Row 12    * * * * * * * * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 13    * * * * * * * * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 14    * * * * * * * * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
Row 15    * * * * * * * * * *  *  *  * * *  * * * *  * * * * * * * * * * * * * * * * * *
* = open seat  X = seat taken
```

Allow the program user to input which seat they would like to reserve. If the seat is taken, they should get a message denying them the seat. If the seat is reserved, the status of the seating should be updated.

Note: In the working application I replaced the untaken seat symbol with a - and the taken seat symbol with a !. This is done to prevent problems with a property called kerning (an X symbol is wider than a * symbol) which distorts the display.

1. Add the following 2D array declaration as a global variable.

```
char [,] seats = new char[15, 30];
```

2. Add the following code to the Form1_Load event handler.

```
string message = "";
for (int r = 1; r <= 15; r++)
        message += "Row " + r + Environment.NewLine;

lblRow.Text = message;
for (int row = 0; row < 15; row++)
        for (int col = 0; col < 30; col++)
                seats[row, col] = '-';
display();
```

3. Add the following code to the display() function

```
string message = "";
for (int row = 0; row < 15; row++)
{
        for (int col = 0; col < 30; col++)
                message += seats[row, col] + "   ";
        message += Environment.NewLine;
}
lblChart.Text = message;
```

4. Add the following code to the btnReserve_Click event handler.

```
int row, col;
//Get row and column from textboxes
row = Convert.ToInt32(txtRow.Text);
col = Convert.ToInt32(txtCol.Text);

//Check to see if seat already reserved
//If already taken display message box and leave function
if (seats[row - 1, col - 1] == '!')
{
        MessageBox.Show("Seat Already taken");
        return;
}
//Assign reserved character to array
seats[row - 1, col - 1] = '!';

//Display the array
display();

//Reset the textboxes and put focus on first textbox
txtCol.Text = "";
txtRow.Text = "";
txtRow.Focus();
```

5. Test your application