

A. Update the Previous Calculator application with Exception handling techniques.

```
private void btnCalculate_Click(object sender, EventArgs e)
{
    try
    {
        string[] input = txtInput.Text.Split(' ');
        if (input.Length != 3)
        {
            throw new ArgumentException("Invalid input format. Please enter two numbers followed by an operator separated by spaces.");
        }

        int num1, num2;
        if (!int.TryParse(input[0], out num1) || !int.TryParse(input[2], out num2))
        {
            throw new FormatException("Invalid number format. Please enter valid integers.");
        }

        char operation = input[1][0];
        int result = 0;
        switch (operation)
        {
            case '+':
                result = num1 + num2;
                break;
            case '-':
                result = num1 - num2;
```

```

        break;
    case 'x':
        result = num1 * num2;
        break;
    case '/':
        if (num2 == 0)
        {
            throw new DivideByZeroException("Cannot divide by zero!");
        }
        result = num1 / num2;
        break;
    default:
        throw new NotSupportedException("Invalid operation!");
}

lblResult.Text = "Result: " + result.ToString();
}
catch (Exception ex)
{
    MessageBox.Show("Error: " + ex.Message);
}
}

```

B. SIGN UP APP

```

using System;

using System.Windows.Forms;

namespace SignUpApp

```

```

{
    public partial class SignUpForm : Form
    {
        public SignUpForm()
        {
            InitializeComponent();
        }

        private void btnSignup_Click(object sender, EventArgs e)
        {
            try
            {
                // Check if passwords match
                if (txtPassword.Text != txtConfirmPassword.Text)
                {
                    throw new Exception("Passwords do not match!");
                }

                // Check if all fields are filled
                if (string.IsNullOrEmpty(txtFirstName.Text) ||
                    string.IsNullOrEmpty(txtLastName.Text) ||
                    string.IsNullOrEmpty(txtDateOfBirth.Text) ||
                    string.IsNullOrEmpty(txtEmail.Text) ||
                    string.IsNullOrEmpty(txtUsername.Text) ||
                    string.IsNullOrEmpty(txtPassword.Text) ||
                    string.IsNullOrEmpty(txtConfirmPassword.Text))
                {
                    throw new Exception("All fields are required!");
                }
            }
        }
    }
}

```

```

        // Check if all fields are strings
        if (!IsString(txtFirstName.Text) ||
            !IsString(txtLastName.Text) ||
            !IsString(txtUsername.Text) ||
            !IsString(txtPassword.Text) ||
            !IsString(txtConfirmPassword.Text))
        {
            throw new Exception("All fields must be strings!");
        }

        // If all conditions met, redirect to next form
        UserDetailsForm userDetailsForm = new UserDetailsForm(txtFirstName.Text,
txtLastName.Text, txtUsername.Text, txtEmail.Text);
        userDetailsForm.Show();
        this.Hide();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message);
    }
}

private void btnReset_Click(object sender, EventArgs e)
{
    // Clear all textboxes
    foreach (Control c in Controls)
    {
        if (c is TextBox)

```

```

        {
            ((TextBox)c).Clear();
        }
    }
}

// Helper method to check if a string contains only alphabetic characters
private bool IsString(string str)
{
    foreach (char c in str)
    {
        if (!char.IsLetter(c) && !char.IsWhiteSpace(c))
        {
            return false;
        }
    }
    return true;
}
}
}

```

In the above code:

- `UserDetailsForm` is the form where user details are displayed.
- The constructor of `UserDetailsForm` accepts parameters for first name, last name, username, and email address.
- `btnSignup_Click` method handles the signup button click event. It checks for password match, empty fields, and whether all fields are strings. If all conditions are met, it redirects to the UserDetailsForm.
- `btnReset_Click` method handles the reset button click event. It clears all textboxes.

- `IsString`` is a helper method that checks if a given string contains only alphabetic characters and whitespaces.