**This assignment is an individual assignment, collaboration is prohibited.**

**You may use your textbook, class handouts, lecture notes, and labs.**

**All code submitted must be your original work.**

* For this assignment you have the freedom to choose your own project with **instructor approval.**
* Your chosen project and feature list must be approved by your instructor.
* Make sure to spend adequate time polishing your app and fixing bugs. Focus on having an app that works well, looks nice, is user-friendly, and has all the important features.
* Consult with your instructor on a regular basis to get input on professionalism and polish.
* There are a few different deliverables required for this project.
  + Project Proposal Document due **April 20th**
    - Must use the provided template.
    - All requested information must be filled out.
    - Must include screen mockups for all features
  + Project Presentation on **May 4th**
    - Slides must be pre-approved and tested by **May 1st**
    - Project Background
    - Explanation of Business Need
    - Purpose of Project
    - Brief Overview of Features
    - Brief Overview of Proposal
    - Live Demo of Website
  + Finished Project uploaded to GitHub by **May 4th**
    - Code must be committed and pushed to git repo by **May 4th** for grading and student must confirm that instructor can download the code
* **Program must implement at least one custom class and object per UML diagram \_\_\_\_\_/200pts**
* Objects must be implemented as a foundational part of your solution

Form2.CS Line 154

* Classes must demonstrate proper encapsulation per the UML diagram

Donor.CS

* Must demonstrate one of the following class relationships
  + Inheritance, Aggregation or Composition
* **Solution must implement programming concepts including**

**variables, custom methods, branching and loops \_\_\_\_\_ / 100pts**

Variables - Form1.CS – Line 88

Custom Method - Form1.CS – Line 26

Branch - Form1.CS – Line 65

Loop - Form1.CS – Line 81

* **Solution must use arrays or collections to store and manipulate data\_\_\_\_\_ / 100pts**
* Must use a List<>   
  Form1.CS - Line18
* **Solution must implement multiple windows\_\_\_\_\_ / 100pts**
* Data must be sent from one window to another  
  Form1.CS - Line100
* **Key Features Implemented from Final Project Proposal: \_\_\_\_\_ / 100pts**
* Register New Users - Form4.CS Line 41
* Store Data - Form4.CS Line 67
* Retrieve Data through Search - Form2.CS Line 154
* Ability to retrieve Donor Emails - Form2.CS Line 20
* Display amounts of different blood types - Form2.CS Line 83
* **Implement Exception Handling for the following scenarios: \_\_\_\_\_ / 100pts**
* **Data Validation on CRUD operations**

Store Data - Form4.CS Line 67

* **Database errors**

Form2.CS – Line 46

* **Microsoft SQL Server Database \_\_\_\_\_/100pts**
* One table limit for data operations.

BloodBank>FinalProject>bin>Debug>**Donors.txt**

**Create**

* Application allows user to insert entities.

Store Data - Form4.CS Line 67

**Search**

* Application allows user to search for entities.

Form2.CS Line 154

**Update**

* Application allows user to search for entities.

Store Data - Form4.CS Line 67

**Delete**

* Application allows user to search for entities.

**Professionalism and Polish \_\_\_\_\_/ 100pts**

* **Most valuable client features have been prioritized and are implemented in a bug-free and usable fashion.** Form1.CS, Form2.CS, Form3.CS, Form4.CS
* Application does not contain any **typos or misspelled words.**
* All error messages are displayed in a user-friendly manner.

Form1.CS – Line 72

* Use the main screen as a menu of sorts to guide users towards the primary functions of the application.

Form1.CS – Line 67(Employee), Line 100(Donor), Line 124(New Donor)