# Project 7 (Optional)

You Must Complete Project 6 First (Optional Project will be graded independently of Project 6 – that means test cases used for testing Project 6 will not be used for this optional project)

# Design Your Own Project 7 (Optional)

- Modify Project 6
  - Think about the employee blood donation scenario
    - What additional functions should be included (at least three) based on the business logic that employees cannot donate more than once in a period of six months, and the employee cannot donate more than once on the same day. Be creative in thinking about more potential functions.
    - Do not change the variable, function, and class names used previously in Project 6.
  - Add 2 additional classes (extending your project 6 code)
    - Illustrate overloading of constructors
    - Illustrate the use of construction delegation
    - Illustrate the use of the friend of a class (it can be for a non-member function, a member function of another class, or another class).

  - Submit all files (.h, .cpp, and main driver – all these files are built by extending your Project 6 program code)
    - Main driver needs to have proper test cases using assert statements.
    - Report.doc or report.txt to explain your business logic, program design, your test cases.
    - Program cannot be compiled and tested using your code will earn zero points.

# Grading will be based on:

- Fully functional program (compiled and run with your header, source, and main driver files)

- Fulfilling these requirements:
  - 3 additional functionalities
  - 2 classes
    - Including an illustration of the use of constructor overload, constructor delegation, and the use of the friend of a class (it can be for a non-member function, a member function of another class, or another class).
      - Zero points for the project if any of these features are missing in your program.
  - 30 test cases (assert statements) in the main driver code.
    - Assert statement must be valid, appropriate, and functional.

- Documentation (report)
  - Description of added functionalities
    - Must include the business logic (in the context of how the business user will use the added features).
  - Program design
    - A thorough explanation of the design of your code.
  - Test cases
    - An explanation of the 30 test cases you used.
      - Why this test case from a code test coverage's perspective?
      - Why this test case from a (business) use case's perspective.

- No late submission will be accepted (period).

- IMPORTANT: zero for code fails to compile ad run
  - Zero for missing the report
  - Zero for missing any components of the report

- Earned grade will be used to replace the lowest project grade.
  - No additional points will be given if the lowest project grade is the same or better than this optional project grade.