CSC 330 – Project 1 (Programming Language C++)

Your deliverables include the following:

Requirements Specification (10 points) – 2-3 pages

Requirements definition is the process of identifying and understanding what the needs of all interested parties are then documenting these needs as written definitions and descriptions. The focus is on *what* problem the system has to solve. The emphasis is on the world in which the system will operate not on the system itself.

The purpose of the *Requirements Specification* step is to capture and express a purely external view of the system. We refer to this view as the public interface of the system. We identify *what* needs to be done starting from the user's needs and requirements. Non-functional specifications also have to be added. We use these to explain constraints such as performance and timing constraints, as well as implementation constraints.

Design Specification (10 points) – 2-3 pages

The purpose of the design phase is to find an appropriate internal architecture for the system that explains *HOW* the requirements are implemented according to an application-oriented viewpoint. The description based on a functional structure and the behavior of each function must be technology-independent. The designer uses the functional design as an entry point for this step.

Our goal is to define or develop the detailed solution to our design problem. In developing the design, we begin with the functional design. We begin by analyzing the problem. Through such analysis, we transform a vague understanding of the requirements into a precise description. The result of such a process is a detailed textual or graphical description of the system. When finished, we have a complete functional definition of the required tasks with no internal contradictions.

As you begin the detailed design phase, structure the system into classes, groups of related classes class or libraries, or types of jobs which include the notions of *helper, supporter,* or *doer*. For each class, specify precisely its operations and relation to other classes. Our goal at this stage is to have the classes crisply defined and the interclass relationships of manageable complexity.

Use UML class diagrams to help formulate the static architecture of your design.

Test Plan (5 points) at least 2 pages

Test Plan identifies *what* tests need to be carried out based upon the original requirements specification. It describes in general terms the following information:

- *What* is to be tested?
- The testing order within each type of test.
- Assumptions made.
- Algorithms that may be used.

Test Cases and Results (5 points) 2-3 pages

The test cases evolved from the test plan, provide the detailed steps for each test.

Source Code (70 Points)

Listings of your program.

NOTE: Your program must be decomposed into main, implementation, and header files.

Your design must include the following

1. The set of rules and constraints under which the journey proceeds.
2. At least 5 or 6 classes.
3. Composition
4. Inheritance and polymorphism

Please, clearly mark in your source code where you are using each of these.