

CS162 Lab 2

Goals

- Identify requirements for a program
- Document the requirements using a simplified class hierarchy
- Develop a simple initial test plan

The purpose of this lab is to develop the habit of understanding the situation before you write code. You will do the analysis for a program, then develop a class hierarchy to document the requirements you identify. This is only design. You will not write any code. That will come in Lab 3.

You will design a program to play a simplified version of war, using dice instead of cards. There will be only **one user**, but **2 “players”** for the game. The user will **indicate the number of rounds to play**. The user will also **specify for each “player” the number of sides on the dice used** and if that player is using **regular or loaded dice**. For example, one player could have a loaded 10-sided die while the other has a 4-sided die. No one said it had to be fair!

To play the game, for each round, you roll a die of the appropriate type for each player. The higher result wins. If the results are equal, it is a draw. The winner of the game is the player who won the most rounds. Your program will print out which player won to the user.

You will design 2 classes, **Die and LoadedDie**. You will use inheritance. You will create a class hierarchy to show that **LoadedDie is derived from Die**. You will also design a **Game** class. Your program will use the Game class to run the game. The Game class will need to keep track of the type of dice for each of the 2 players, the number of rounds to play, and some way to maintain the score. The Game class will not be part of an “is-a” relation, but a “has-a” relation. Therefore, you should have at least 3 classes for this lab. Do you NEED a player class? Think about it.

In your design, you should list your requirements. You should clearly indicate which nouns are classes, which nouns are data members for which classes, and which activities are members of which classes. You should describe any actions required. You should include an outline of the design for your main program, indicating how it will use the Game and Die classes.

You also need to include a class hierarchy diagram displaying the organization and/or relationship of the different classes. Clearly indicate any **“is-a” and “has-a” relations**. The box for each class should **separate name, data, and function members**. An example is included at the end of this file. If you do not want to struggle with a drawing program, please draw it neatly by hand and scan it. Notice that it is not detailed. The goal is to better understand the problem.

At last, you will also **create a draft test plan**. Thinking about testing while doing your design can help spot problems. For the test plan: what options are available to the user? What results

would you expect for each option or combination of options? Check out the test plan example in the reading materials.

Grading

- English writing (is it clear?) – 10%
- Clearly state the requirements for the program – 30%
- The class definitions provide all required behaviors – 20%
- The class hierarchy (including the diagram) clearly reflects the requirements – 20%
- The initial test plan covers the current design – 20%

HINT: This lab is mostly writing (an important skill for the workplace), so treat it as a paper. Draft it, then ask someone to read it. They do not need to be a computer person! You want to know if it makes sense.

Example class hierarchy diagram:

