Our code consists of five main classes: Simulation, Store, Tool, Customer, and Rental. At the highest level, Simulation represents the initialization of the store and customers as well as the passage of time. Store represents the store rented from. It handles creating and tracking its inventory of tools, checking if tools are available, tracking money, recording current and old rentals, and processing incoming rentals. Customer represents the customer renting from the store. It handles tracking how many tools and for how long the customer rents, creating a rental, and tracking the rentals made by that customer. Customer has 3 subclasses, one for each type of customer. Each subclass defines the range of possible tools rented and possible days in the rental, while the superclass Customer has a method to generate the actual number for each based off any range. Rental represents a rental of tools. It is responsible for tracking what customer made it, how many and which tools are involved, the total cost of the rental, and the total days and remaining days in the rental. Tool represents a tool rented in the store. It is responsible for having a price and an availability flag. Tool has subclasses for each type of tool, making 5 total. Each subclass specifies the average price of that tool, while the top class has a method to generate the price given any average price. Both Customer and Tool use the strategy pattern; only their subclasses should be instantiated. The code fails if either a Customer or Tool is instantiated (instead of a CasualCustomer or PaintingTool, for example).

The general structure looks like this: Simulation starts the simulation by creating the store and customers. Each Customer when created decides how many tools it will rent for how many days if it is chosen. Store then creates the tools, 4 of each of the 5 types. Each Tool generates its exact price and unique name, both based off what type of tool it is. The first day then happens, with a random number and selection of Customer classes being chosen by Simulation to rent that day. The selected Customers one by one check if they can rent the

desired number of tools (without having more than 3 tools), ask the Store if there are enough tools to rent and, if so, which ones are available. It then creates a Rental object with all the necessary info. Finally, it sends this Rental to the Store. The Store takes the rental and processes it, making the involved tools unavailable and adding the money to the count. Store adds the rental to current rentals and waits for the next customers. Once all customers have submitted their rentals or skipped making one (if the store didn't have tools available), the Simulation moves to the next day. Before the day starts, all current rentals have their 'days remaining' counter decremented. If any hit zero, the store frees the tools and moves the rental to the old rental list and the customer lowers the amount of tools it currently has rented to match. This process repeats in a day->pre-day->day cycle until after the 35th day. At that point, the Simulation tells Store to print the report.