

Due Dates:

1. Jun 5..... Proposal Due
2. Jun 28..... Object Diagrams Due
3. Jul 17..... Final Program Due

You must choose your own project. You are strongly encouraged to use C++. You are also encouraged to use the Windows 9x,2000,NT platform. HOWEVER, you may use ANY object-oriented language, and ANY operating system, with the only requirement being that the operating system and language you use must be available in an open-use laboratory here on campus. It is necessary that we be able to compile and run your code as part of the grading process.

You must create an object-oriented model of something. The model is more important than the actual functioning of the program. Your model must be comprehensive, and your program must do something meaningful using the capabilities of the model. Your proposal must be approved, by me, before you can proceed with the rest of the project. Failure to submit ANY PART of the project will result in a zero grade for the entire project. DO NOT ASSUME THAT I WILL AUTOMATICALLY APPROVE YOUR PROJECT! It is likely that 25-50% of the projects will be rejected outright, or returned for more detail before they are approved.

Here are some sample projects.

1. Construct a model of a bee-hive. The program will provide a simulation of one summer, including collecting the nectar, and performing the harvest.
2. Construct a model of a city with 16 blocks arranged in a 4x4 array. The model will include traffic, stop lights, and pedestrian traffic. The program will demonstrate the effect on traffic of a buy-one-get-one-free sale at the local video store.
3. Construct a model of an electric train set. The program will permit the user to move trains around the track, and will provide an animation of the trains.
4. Construct a model of a public library, its holdings, its customers, and its checkout process. The program will be a user interface into the holdings, and a user interface that allows librarians to check books and other materials out to the customers.

The Proposal must contain a description of the thing you are going to model. **DO NOT** describe a program. A database application **IS NOT ACCEPTABLE!** You must convince me that your model is complicated enough to warrant an object oriented design, rich enough to support at least one reasonable application, and simple enough to be completed in one semester. **NO HANDWRITTEN PROPOSALS WILL BE ACCEPTED!** I repeat: **PROPOSALS MUST BE TYPEWRITTEN!** Proposals must be turned in by you personally, and must be on paper. E-Mail is not acceptable. Don't ask a friend to turn it in for you. If you have a valid excuse for missing class the day the proposal is due, turn it in early, or **contact me beforehand** to make alternative arrangements.

The object diagrams must use the BOOCH notation, as presented on the inside cover of the book. Diagrams should be constructed using computer graphics tools. Handwritten material is not acceptable. Again, it must be on paper, turned in by you personally.

The final assignment must include the following items.

1. A booklet **ON PAPER** containing
 - a. Your proposal, corrected if necessary
 - b. Your object diagrams
 - c. Module diagrams
 - d. A verbal description of the code and what it does (1-2 pages)
 - e. A user manual
 - f. A printout of your source code. Do not include code that is generated by the design system or by tools such as YACC and LEX.
2. A diskette or a CD-ROM containing the following
 - a. Your source code. If you use a PC development system, include the entire project folder. Otherwise, include all source files.
 - b. A running version of your program with all required run-time libraries. The user manual in the booklet must be sufficiently detailed to allow us to test your program to guarantee that it works.

A perfect score on the assignment is 15. You will receive 5 points for your object diagrams (assuming they are perfect), and 10 points for the program. The 10 points for the program will be broken down as follows.

3 points - Program compiles, produces a valid copy of the running program.

3 points - The program runs correctly.

3 points - Module diagrams, user manual and code description are present and accurate.

1 point - Proposal and object diagrams present in the final booklet.

The Shopping Center

Build a model of a strip shopping center. For simplicity, assume that all stores are owned by the shopping center, and that each is run by a manager who is employed by the shopping center. Each store is required to pay rent out of its profits, based on the size of the store. Each store is a concession like McDonald's, and must pay a concession fee out of its profits. There are 15 stores in the center of various sizes, however your software should be usable for other shopping centers of other sizes. For simplicity, assume that the stores are arranged in a single line. Also, for simplicity, assume that each store stocks no more than three different products, although your software should allow for an unlimited number of products. Each store keeps an inventory of products along with their names, current quantity, and prices. Each store has several employees. For simplicity, you may assume that there are no more than three employees per store, but your software must allow for an unlimited number of employees. Each employee has a salary, and a history of payments, including taxes, other deductions, and net salary. Each store has a number of available positions. An employee must be hired into a position. Each position has a category, a base salary, and an annual-increase percentage. The manager of a store is a position. As in any shopping center, stores open and close, and change from one type to another. Each store has a current type, and an inventory and position list that corresponds to the type of store. Different types of stores have different types of inventory requirements, for example a shoe store may get new inventory once a month, while a restaurant needs to order fresh fish every day. Each store needs to keep track of its sales. For simplicity, do not model individual purchases, only daily sales. (It should be possible to expand your model to include individual sales.)

Your model must be able to support the following tasks.

1. Make a map of all the stores, and the name of the concession in each. Leave empty stores blank.
2. Give total profits for each store on a monthly basis.
3. Give total profits for the shopping center on a monthly basis.
4. List the employees of a store and their salaries.
5. List unfilled positions for each store.
6. List inventory, and indicate when purchases are necessary.
7. Other cool stuff.

The Library

Create a model of a library, keeping the following considerations in mind. The library has several departments, Reference Books, Circulating Books, Periodicals, Video Tapes, Audio Materials and Maps. Each department has its own lending policy. For example, books may be checked out for four weeks, video tapes for four days. This system needs to be able to locate each item within the library, tell whether it has been checked out, which librarian checked it out, who checked it out, whether it is overdue, or lost. The system should include library hours, and librarian work schedules.