Advanced Programming COSC 1295

Project Discussion

Sherri McRae
s3117889

Our original plan included a Menu class with the menu items linked to methods in a Driver class which in turn called on a Person class. The Person class had two subclasses: Adult and Child.

Although this basic design has remained the same, a new class, the "GlobalClass" was created to store a number of static variables that were used many times: those that represented menu options.

An ArrayList was used to store the objects of Adult and Child. I came across the ArrayList class while researching ways to get around the need for a standard one-dimensional array which requires a size to be defined upon initialization. While we could have set an array at a particular size, we felt that it would take up unnecessary memory space, especially if we anticipated a large number of profiles. We felt it was better for the collection of profiles to expand as needed.

Initially, the Person class was not an abstract class, but it made sense to declare it as such, if only to highlight that the program would not be creating instances of Person. Only Adult classes and Child classes could be used to create objects (in this case profiles) to store in the array.

The idea of using interfaces was explored in the design phase of this project. However, it seemed as though interfaces would complicate the design. The possibility of Adult/ Child interfaces was discussed but the strong "is-a" relationship between Person/ Child and Person/ Adult made the superclass/ subclass relationship the simplest way to take advantage of inheritance and polymorphism.

While the sections of code required to add friends, create profiles, delete friends and so on was easy to write, the trickiest part the assignment, at least for me, was figuring out how to locate a particular profile in the array of profiles. The exact location of each profile would not be known to a user. The method getIndexByProperty() was developed for that purpose. It is a simple program that takes a name, as a string, and returns the index as an integer. Once it was possible to locate an object, the other specifications were fairly easy to manage.