

CSCI-1200 Computer Science II — Spring 2006

Homework 4 — Wish Lists

Write a program that a store could use to keep track of wish lists for gifts. An individual can register for an item that he or she would like, and an individual can purchase an item for a different individual. This could be used for weddings, birthdays, or holiday gifts. *Please carefully read the entire assignment before beginning your implementation.*

The input for the program will come from a file and the output will also go to a file. These file names are specified by command-line arguments. Here's an example of how your program will be called:

```
wishlist.exe requests.txt results.txt
```

The form of the input is relatively simple. Each line of input specifies a separate request. There are four types of requests, described below. You may assume the input file strictly follows this format (i.e., you don't need to worry about format error-checking).

```
a Bob hammer 10.50
p Bob hammer Alice
w Bob
b Bob
```

1. The first request, indicated by the letter 'a', says to add a particular item (**hammer**) and its price (**10.50**) to the wish list for the individual (**Bob**). If the individual does not yet have a wish list, a new wish list must be added before the item is added to the wish list. Do not bother to check if an item is already on the wish list, multiple items with the same name may be added to the wish list and later purchased. The output for the first request is:

```
New wish list added for Bob
```

when a new wish list is created and:

```
Item hammer added to wish list for Bob
```

for adding the item.

2. The second request, indicated by the letter 'p', says that **Alice** would like to buy item **hammer** for **Bob** from **Bob's** wish list. Your program should check first to see if **Bob** has a wish list, outputting an error message if not. The form of the output should be:

```
Error: no wish list for Bob
```

If **Bob** has a wish list, the program should record the purchase. If the item is not on the wish list, an error message should be output. The form of this output should be:

```
Error: hammer is not on the wish list for Bob
```

Otherwise, the item should be marked as purchased by **Alice** and an output message should be issued. The form of this output should be:

```
Alice bought hammer for Bob
```

3. The third request, indicated by the letter 'w', asks for the wish list for the individual **Bob**. If the individual does not have a wish list, an error message of the form:

```
Error: no wish list for Bob
```

must be output. Otherwise, the wish list of items not yet purchased must be output. The output must start with:

```
Items on wish list for Bob
```

and then an output line for each item with the name and the price (with a \$ in front). Use iomanipulators to format the price so 2 decimal places are printed. Note that if an item has been purchased, it should not appear on the output wish list. If multiple instances of an item were added to the wish list and only some of them have been purchased, the remainder should be listed with this output. The items should be listed in order by price, with the most expensive item first. If no items remain on the wish list for this individual, output:

```
-none-
```

4. The fourth request, indicated by the letter 'b', asks for the items from the wish list for individual Bob that have been bought. If the individual does not have a wish list, an error message of the form:

```
Error: no wish list for Bob
```

should be output. Otherwise, the list of items that have been purchased must be output. The output must start with:

```
Items purchased for Bob
```

And be followed by a line of output for each item purchased in this form:

```
hammer bought by Alice
```

The purchased items should be ordered alphabetically by the name of the purchaser (to simplify the writing of "thank you" notes). If no items were purchased for this individual, output:

```
-none-
```

Sample input and output files are posted on the course web site. Please follow these examples closely.

Additional Requirements, Hints and Suggestions

- **You may not use vectors or arrays for this assignment.** Use standard library lists instead.
- You must write at least one new class. For example, you probably want to have a `WishList` class that does the work of maintaining a wish list for one individual. You may write additional classes.
- We have provided a partial implementation of the main program to get you started. You may use none, a little, or all of this, as you choose, but we strongly urge you to examine it carefully. There are member-function calls to our version of the `WishList` class, so you can deduce how some of the member functions in our solution work.
- Do all of your work in a new folder named `hw4` inside of your CSII homeworks directory. Use good coding style when you design and implement your program. Be sure to make up new test cases and don't forget to comment your code! If you have any notes you want the grader to read, write them in a *plain text* file named `README.txt`. When you are finished please zip up your `hw4` folder exactly as instructed for the previous assignments and submit it through the course webpage.