

**Due Date:**

Monday, March 17, 2014

**Delivery**

This is a multiple file program that must be archived using the **tar** command. Deliver a single file named `pa3.tar` (all lowercase) as an **attachment** to an email that you send to the class account `csc3320@orion.ucdenver.edu`. Put PA3 in the subject field, and your name in the body of the email. Here is a tar command for this assignment.

```
tar -cvf pa3.tar pa3.cpp pa3templates.h pa3templates.tem pa3functions.h  
pa3functions.cpp table.h table.tem
```

**Program objectives**

The objectives of this assignment are as follows.

An ability to use current techniques, skills, and tools necessary for computing practice (ABET i).

**Value**

This program is worth 15 points. The distribution of points will be as follows.

Criterion	Value
Templates	1
Hash table class	2
Hash function	1
Program style (includes globals)	2
Correct output with annotation	6
Error checking	1
Exception	1
Struct	1

**Problem**

You are to create a simple database using a hash table. The database will keep track of a small club consisting of members who pay monthly dues. The dues are collected each month and are based on the members' participation in volunteer activity. The more a member participates in monthly activities, the less the dues for that month. Use the database to produce a yearly report for each member, listing their monthly dues.

**Input**

A file containing a list of member names and their monthly volunteer time in hours. The file name should be read from the command line in `argv[1]`.

**Output**

A table listing the member, the months (1 year, 12 months), and the dues for each month. This table is to be displayed on the standard output (see below).

**Class requirements**

Your hash table class must be based on a static array which is dimensioned by a non-type parameter. In addition, the template for the class should list two other parameters—a type for the table and a type for a pointer to a hash function. Furthermore, the class must contain at least the following members.

1. At least one constructor that takes the parameter representing a pointer to a hash function.
2. Insert function to insert something into the hash table.
3. A display function shows the hash table in a formatted tabular form on the standard output.
4. A nested exception class that throws an error when the capacity of the table is exceeded. This class should be derived from `std::exception` and should be named `table_overflow`. The exception should print a string “Overflow error”, when it is caught. Use a reference to the base class as the catch argument.

**Program requirements**

1. Present the user with a greeting.
2. Read the file with the members and the hours in `argv[1]`.
3. Build the hash table and compute the statistics.
4. Display the hash table.
5. Don’t forget to enable exception handling in the program with `try/catch/throw`.

**Notes**

1. To facilitate this project, create a struct that stores the member name, hours worked for each month, and monthly dues for each month. You may add other data members if you like.
2. Your hash function should compute a table slot based on a person's name as follows. Table slot = key mod table\_size, where key is computed as the sum of the ASCII values of the person's name. If there is a “collision” (table slot is already occupied), use the following function to resolve it.

Table slot = (key +  $i$ ) mod table\_size, where  $i = 0, 1, 2, \dots, \text{table\_size}-1$

3. The format of the input file will be as follows.

Amy 7 4 9 8 9 5 5 3 7 10 9 10

Bob 3 9 10 5 8 9 9 10 4 7 9 10

// Other entries

The space-delimited numbers represent hours worked each month beginning with January and ending with December.

// Output

Amy

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

5.50 4.00 7.50 2.00 5.50 10.00 8.50 7.50 6.00 6.50 8.00 5.50

Bob

// Same as above

4. Compute the dues (hours worked \* rate), where the rates are as follows (in dollars)
- |               |       |
|---------------|-------|
| hr>=0 && hr<2 | 10.17 |
| hr>=2 && hr<4 | 8.15  |
| hr>=4 && hr<6 | 6.13  |
| hr>=6 && hr<8 | 4.11  |
| hr>=8         | 2.09  |