IICS

FACULTY OF INFORMATICS

University of Wollongong

CSCI124

July Session 2014

Assignment 2

*(Individual Work – 6% of subject marks)*

**Objective:**

In this assignment, you will learn to write programs in multiple source files with binary file processing.

**Remember that:**

**1. All programs should be able to run on the lab’s computers.**

**2. You must put the following information on the header of each text and source file you will be submitting in this assignment:**

**Student’s full name:**

**Student’s ID:**

**Modification Date:**

**Purpose of this file (or program):**

**3. Assignments that are not able to be compiled will result in zero mark given to the assignment.**

**4. You must only use the C++ features that have already been covered in the lectures**

Problem Specification:

A bank would like to have a program to process the customer’s bank loans information. Each customer at the bank is described by their first name, last name, customer id, amount of loan, duration of loan in months, interest rate for loan, and total paid. The program should store the customer’s records in a binary file and supports a variety of operations such as add a new customer record, modify customer records, search and display customer records, etc.

Your task is to implement this according to the specification provided below:

You are to use a binary file to keep the records of all customers. Identify and define a suitable structure to keep your customer record.

The program should be able to perform the following functions:

1. A new customer can be added to the database – when this occurs the user will be prompted to enter the various details that describe the customer. Your program should be able to perform error checking on the input provided by the user. The customer id should be unique.
2. The program should be able to display:
   1. A particular customer’s information together with the outstanding balance to be paid.
   2. ALL customers’ information that have finished paying the loan amount.
   3. ALL customers’ information that still have outstanding amount to be paid.
3. Update a customer’s record by adding a new amount (the monthly payment) to the total paid indicating that the customer has done payment. When this option is selected, the user should be prompted to enter the customer id. Search and display the customer information and also the monthly payment due. The monthly payment is calculated as:

Monthly payment =

where,

If the user enters a value that is less than the monthly payment, display a message indicating the situation and continue to update the record. Negative amount should be regarded as invalid.

1. Delete a customer’s record in the database. The user should be prompted to enter the customer id. If the record exists, it should be displayed. The user should then confirm if the record should/ shouldn’t be deleted. Because the search is done on a unique attribute you will never have multiple records which are candidates for deletion.

Your program is NOT permitted to store any customer records in memory, all operations must be performed by reading/ writing directly to/ from the file. Where appropriate a temporary buffer may be used to store an individual record.

You will also need to consider how deletion should be implemented. (HINT you may consider marking records as those ready for deletion and overwrite them when subsequent additions are done).

The code should be written in multiple appropriate header files and cpp files.

A suitable driver should be defined in main.cpp which displays a menu system, allowing the user to select from the above options. Include also an option to terminate from the program. When invoked the program should prompt the user to enter the filename of the binary file to be used.

**Assessment Criteria:**

|  |  |
| --- | --- |
| **Assessment Criteria** | **Marks Allocated** |
| Correctness | 1 |
| Coding | 3 |
| Readability and Documentation | 1 |
| Output (clear and well formatted) | 1 |
|  |  |
| **Total** | **6** |

Submission:

You are to submit the softcopy of the project containing all files (.cpp, .h, and sample text files used) to Moodle.

The completed Assignment must be submitted latest by **Monday, 29th September 2014 , at 5 pm**.

Late submissions will be marked with a 25% deduction for each day.

Submissions more than three days late will not be marked, unless an extension has been granted.