General

The project was evaluated in terms of the following criteria:

Correctness (50 points)

See “Correctness Evaluation” below.

Software Design Document (25 points)

- The project document was graded with one of the points (25, 18, 10, 0) which correspond to good, fair, poor, and none respectively.
- The document should include the sections (all or those that are sufficient to explain the project) as stated in the Programming Project Documentation.
- The sections should include sufficient information and be written in a clear way such that the document serves as both a user manual and a technical manual. In other words, when reading the document, a user should be able to use the program and a programmer should understand the code without any difficulty.

Software Quality (25 points)

- The software quality was graded with one of the points (25, 20, 15, 10, 0) which correspond to very good, good, fair, poor, and very bad respectively.
- The program should be written in a modular way. For instance, if a task you want to do is not trivial, then probably you should use some auxiliary functions.
- More complex functions should be written in terms of the already written simpler functions.
- The code should be highly commented. Marks were deducted for missing comments.
- The filenames should be the same with the ones stated in the project specification.
- Object-oriented approach should be preferred where applicable.

Correctness Evaluation

For each question, you got the points shown below if the program returned the correct answer. If your code took any more that 10 seconds to run for a question, the execution was aborted and the program was accepted as unsuccessful for that question. The cases were carefully selected so that in all situations, correct code would finish within a second or two. For each function, the test cases, the expected answer, and the points are as follows:

Test Case 1: Insert Member → umut, swimming, 9, 2009
Expected Answer: successful insertion
Points: 3

Test Case 2: Insert Member → ismail, swimming, 10, 2012
Expected Answer: successful insertion
Points: 3

Test Case 3: Insert Member → ismail, swimming, 10, 2004 → then change the name to ali
Expected Answer: duplicate name error then successful insertion
Points: 3

Test Case 4: Insert Member → alp, basketball, 10, 2010
Expected Answer: successful insertion
Points: 3

**Test Case 5:** Find Preferred Sports  
**Expected Answer:** none  
**Points:** 5

**Test Case 6:** Insert Member → bulk addition (5 different people (a,aa,aaa,aaaa,aaaaa) to football, 2 different people (b,bb) to swimming)  
**Expected Answer:** successful insertions  
**Points:** 3

**Test Case 7:** Find Preferred Sports  
**Expected Answer:** football  
**Points:** 5

**Test Case 8:** Update → alp, basketball, 10, 2011  
**Expected Answer:** successful update  
**Points:** 5

**Test Case 9:** Update → ali, basketball, 10, 2011  
**Expected Answer:** ERROR since 4 months passed since 10, 2004  
**Points:** 5

**Test Case 10:** Delete Inactive  
**Expected Answer:** ali and umut are deleted  
**Points:** 5

Close the program and launch it back to test whether it successfully saves and reads the database file.

**Test Case 11:** Insert Member → ali, basketball, 10, 2011  
**Expected Answer:** successfully added  
**Points:** 5

**Test Case 12:** Insert Member → ismail, football, 10, 2014 → then change the name to ayse  
**Expected Answer:** duplicate name error then successful inserton  
**Points:** 5

**TOTAL:** 50