# Exercise 1: User Database

Create a User class to hold information about a single user:

* Last name
* First name
* Social security number
* Age
* A tagline (favorite quote, saying, anything)

The user class should have stream operators to read it and write it from/to streams of bytes. Test it with “cin” and “cout”. Provide accessor methods as desired.

Create a UserDB class to hold an array of Users. You can set the database to hold a maximum of 100 users or you can make it dynamically sized. The class should have stream operators that read the entire database in and out of a stream of bytes. You can put the size of the database as the first entry in the byte stream.

Create a “test” program that is an infinite input loop for user interaction. The commands could include:

1. Read database from a file
2. Write database to a file
3. Print database on screen
4. Add a new entry to the database (read entry from keyboard)

### User.h

#ifndef USER\_H\_

#define USER\_H\_

#include <ostream>

#include <istream>

#include <string>

class User {

// Make the "operator>>" a friend so the function can get access to private parts

friend std::istream & operator>>(std::istream & os, User & user);

private:

std::string firstName;

std::string lastName;

int age;

std::string ssn;

std::string tagLine;

public:

// Note the default argument values

User(std::string first="", std::string last="", int age=0, std::string ssn="",

std::string tagLine="");

// Getters

std::string getFirstName();

std::string getLastName();

int getAge();

std::string getSSN();

std::string getTagLine();

// Two setters ... just for demo

void setAge(int newAge);

void setTagLine(std::string newTagLine);

};

// Stream operator functions

std::ostream & operator<<(std::ostream & os, User & user);

std::istream & operator>>(std::istream & os, User & user);

#endif

### User.cpp

#include "User.h"

#include <string>

using namespace std;

#include <ostream>

#include <istream>

User::User(string \_firstName, string \_lastName, int \_age, string \_ssn, string \_tagLine)

{

firstName = \_firstName;

lastName = \_lastName;

age = \_age;

ssn = \_ssn;

tagLine = \_tagLine;

}

string User::getFirstName()

{

return firstName;

}

string User::getLastName()

{

return lastName;

}

int User::getAge()

{

return age;

}

string User::getSSN()

{

return ssn;

}

string User::getTagLine()

{

return tagLine;

}

void User::setAge(int newAge)

{

age = newAge;

}

void User::setTagLine(std::string newTagLine)

{

tagLine = newTagLine;

}

ostream & operator<<(ostream & os, User & user)

{

os << user.getLastName() << " " << user.getFirstName() << " " << user.getAge()

<< " " << user.getSSN() << user.getTagLine() << endl;

return os;

}

istream & operator>>(istream & is, User & user)

{

char tag[256];

is >> user.lastName >> user.firstName >> user.age >> user.ssn;

is.getline(tag,256);

user.tagLine = tag;

return is;

}

### UserDB.h

#ifndef USER\_DB\_H\_

#define USER\_DB\_H\_

#include "User.h"

#include <ostream>

#include <istream>

class UserDB {

private:

User\* users; // Dynamic memory here ... a maximum

int numberOfUsers; // Current number of users

public:

UserDB();

int getNumberOfUsers();

User getUserAt(int index);

void addUser(User u);

};

// No need for "friends" here.

std::ostream& operator<<(std::ostream& os, UserDB& db);

std::istream& operator>>(std::istream& is, UserDB& db);

#endif

### UserDB.cpp

#include "UserDB.h"

#include <ostream>

#include <istream>

#include "User.h"

using namespace std;

UserDB::UserDB()

{

users = new User[100];

numberOfUsers = 0;

}

int UserDB::getNumberOfUsers()

{

return numberOfUsers;

}

User UserDB::getUserAt(int index)

{

return users[index];

}

void UserDB::addUser(User u)

{

users[numberOfUsers] = u;

++numberOfUsers;

}

ostream& operator<<(ostream& os, UserDB& db)

{

os << db.getNumberOfUsers() << endl;

for(int x=0;x<db.getNumberOfUsers();++x) {

os << db.getUserAt(x);

}

return os;

}

istream& operator>>(istream& is, UserDB& db)

{

int x;

is >> x;

for(int y=0;y<x;++y) {

User u;

is >> u;

db.addUser(u);

}

return is;

}

### db.cpp

#include <iostream>

using namespace std;

#include <fstream>

#include "User.h"

#include "UserDB.h"

void main() {

UserDB db;

int inp;

fstream myDbFile;

string filename;

User newUser;

while(true) {

cout << "Commands:" << endl

<< " 1 fname (read from fname)" << endl

<< " 2 fname (write to fname)" << endl

<< " 3 last first age ssn tag (create new)" << endl

<< " 4 (print db)" << endl;

cin >> inp;

switch(inp) {

case 1: // Read from file

cin >> filename;

myDbFile.open(filename, fstream::in);

myDbFile >> db;

myDbFile.close();

break;

case 2: // Write to file

cin >> filename;

myDbFile.open(filename, fstream::out);

myDbFile << db;

myDbFile.flush();

myDbFile.close();

break;

case 3: // Create new user

cin >> newUser;

db.addUser(newUser);

break;

case 4: // Print DB

cout << db;

break;

default:

return;

}

}

}