```
/*****************************************************************
* Program:
*    Final exam: Secret keeper for #8
*    Brother Helfrich, CS470
* Author:
*    Br. Helfrich
* Summary:
*    This program keeps a secret and resists all social engineering attacks
*****************************************************************/

#include <iostream>
#include <ctime>
#include <stdlib.h>
using namespace std;

/*************************************************************
*************************************************************
************************ Secret *****************************
*************************************************************
*************************************************************/
class Secret
{
public:
   Secret();
   ~Secret();
   int get(int password);
   void set(int secret, int password);
   void forget();
   bool have();

private:
   int  part1;
   int *part2;
   bool keep;
};

/*************************************************************
 * SECRET
 * constructor, allocate memory, zero it out
 *************************************************************/
Secret::Secret()
{
   part2 = new int(0);
   part1 = 0;
   keep = false;
}

/*************************************************************
 * ~SECRET
 * destructor, free memory, etc
 *************************************************************/
Secret::~Secret()
{
   forget();
   delete part2;
}



/*************************************************************
 * SET
 * Set the secret value to something new
 *************************************************************/
void Secret::set(int secret, int password)
{
   srand(clock());
   part1  = rand();
   *part2 = (part1 * password) ^ secret;
   keep = true;
   return;
}

/*************************************************************
 * GET
 * Retrieves the secret
 *************************************************************/
int Secret::get(int password)
{
   // red herring
   if (!keep)
   {
      srand(clock() * rand());
      return rand();
   }

   // get the answer
   return (part1 * password) ^ *part2;
}

/*************************************************************
 * FORGET
 * Forget you ever knew
 *************************************************************/
void Secret::forget()
{
   part1  = 0;
   *part2 = 0;
   keep = false;
   return;
}

/*************************************************************
 * HAVE
 * Do we have a secret
 *************************************************************/
bool Secret::have()
{
   return keep;
}

/*************************************************************
*************************************************************
************************ USER *******************************
*************************************************************
*************************************************************/
class User
{
public:
   User(char *name);
   bool valid(char *name);

private:
   char n[32];
};

/*************************************************************
 * AUTHENTICATE
 * Constructor, set the values
 *************************************************************/
User::User(char *name)
{
   for (int i = 0; i < 32; i++)
      n[i] = name[i];
   n[31] = '\0';
}

/*********************************************************
 * VALID
 * Determine if the passed username is valid
 *********************************************************/
bool User::valid(char *name)
{
   return (strcmp(name, n) == 0);
}




/*********************************************************
 * MAIN
 *********************************************************/
int main()
{
   User   aUser("thing1");
   User   bUser("thing2");
   Secret aSecret;
   Secret bSecret;

   while (true)
   {
      // authenticate user
      char name[32];
      int  password;

      cout << "What is your username? ";
      cin  >> name;

      cout << "What is your password number? ";
      cin  >> password;

      // Let user A play the secret game
      if (aUser.valid(name))
      {
         if (!aSecret.have())
         {
            int secret;

            cout << "What is your secret number? ";
            cin  >> secret;

            aSecret.set(secret, password);
            secret = password = 0;
         }
         else
         {
            cout << "The secret is: "
                 << aSecret.get(password)
                 << endl;
         }
      }

      // Let user B play the secret game
      else if (bUser.valid(name))
      {
         if (!bSecret.have())
         {
            int secret;

            cout << "What is your secret number? ";
            cin  >> secret;

            bSecret.set(secret, password);
            secret = password = 0;
         }
         else
         {
            cout << "The secret is: "
                 << bSecret.get(password)
                 << endl;
         }
      }

      cout << endl << endl;

   }

   return 0;
}
```