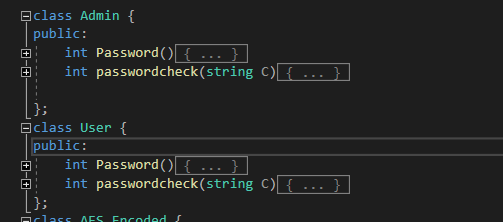
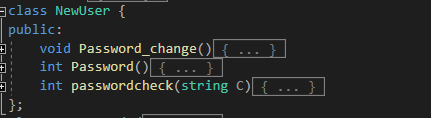
***Uzair Asif***   
***Abdullah Malik***

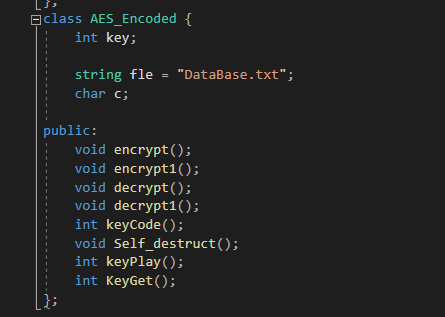
**Attention:**

**C & C++ has been used.**  
**Any text files within Network Security A1 is advised not to be deleted.**  
  
**Functionality of Code:**  


**Classes Admin and User:**Admin & User classes are called and used to call in function within menu to take passwords and check if password that is entered is correct or not. If password is entered incorrectly then it is re-entered.  
Moreover. the password that is entered is taken in form of “\*\*\*\*\*\*”.

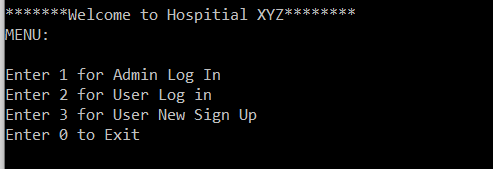


**Class NewUser:**Class calls in functions for the registering new users. This uses File handling to save passwords.

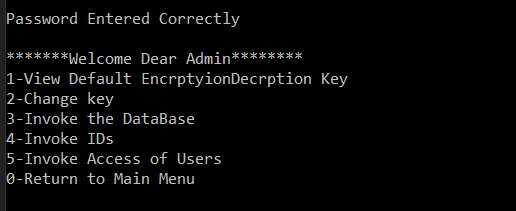
****

**Inheritance:**  
CLASS ASCII\_Encoded is used to call in all the functions in its Child classes.  
Simple Encryption Method is used. A key is used to encrypt the file and decrypt it.  
void encrypt(); uses default key to encrypt the file.  
void decrypt(); uses default key to decrypt the file.  
int keyCode(); is used to Enter the new key within the admin log and is called by   
int keyplay (); which returns the key into another function void encrypt1();.  
void encrypt1(); does not uses default but instead requires the key from the admin to encrypt the file.  
int KeyGet(); is a getter function for the void decrypt1() function which uses it to decrypt for the encryption done by void encrypt1();  
Void Self\_destruct(); is called by the admin to destroy the DataBase And make Encrypted versions of it. Also Used to destroy the IDs or Any given Files.  
Can be used to regain access for users.  
Void Display\_File(); is for default version of Encrypted File while Void Display\_File1() is used for Key given by admin. Both displays the Decrypted files.  
int Display\_ID(int F); is used to read from IDs.Text and Display Text File.

**Menus:**

****

This is the main menu for the program. User Log in is simple as it calls in functions to display the decrypted files and is used to verify if user knows the password or not while Admin Login-In have more Functions. User Sign-Up is used for creating new log for new patient.



Admin pops up once User enters 1. User Before menu is displayed, password is required to enter into menu of admin. Admin can see the default keys for encryption and decryption. He can also change the keys within the program and can delete the DataBase and IDs. He can also deny Users from accessing Users Log-In.

**Note1:**  
System (clsr) is often used in the program to give a neat look to Output Prompt.  
Sleep (1000000) is used to delay the program for 3 seconds.

**Vulnerabilities:**Default Encryption can easily be obtained through Code.

Weak passwords always play a major role in any hack. For the ease of user, sometime applications do not enforce password complexity and as a result of that users use simple passwords such as password, password123, Password@123, 12345, god, own mobile number etc. Weak password does not always mean length and the characters used; it also means the guess ability. Name@12345, it looks quite complex password but can be guessable. So do not use password related to name, place, or mobile number. Weak passwords can be guessable or attacker can brute force if the length of the password is very small, so try to use random strings with special characters. Though that can be hard to remember as a security point of view it’s quite secure.

**Prevention:** Strong password is needed.   
Admin is not well hidden within the code and it can be obtained. Gaining access to Admin can lead to Key for Encryption and Decryption. Authorization failure. It means that when a function is called on the server, proper authorization was not performed. A lot of times, developers rely on the fact that the server side generated the UI and they think that the functionality that is not supplied by the server cannot be accessed by the client. It is not as simple as that, as an attacker can always forge requests to the “hidden” functionality and will not be deterred by the fact that the UI doesn’t make this functionality easily accessible. Imagine there’s an admin panel, and the button is only present in the UI if the user is actually an admin. Nothing keeps an attacker from discovering this functionality and misusing it if authorization is missing.

**Prevention:** On the server side, authorization must always be done. Yes, always. No exceptions or vulnerabilities will result in serious problems.  
Algorithm for Keys can easily be deciphered. So far, we have been discussing symmetric key algorithms such as AES, HMAC, CMAC, GCM, and CCM. These algorithms are known as symmetric (or shared secret) algorithms, since all parties share the same key values. Revealing this key would compromise the security of the system.  
 Buffer overflows can lead the program to crash as the size isn’t handled well enough. Buffer overflows can affect all types of software. They typically result from malformed inputs or failure to allocate enough space for the buffer. If the transaction overwrites executable code, it can cause the program to behave unpredictably and generate incorrect results, memory access errors, or crashes.

**Prevention:** The easiest way to prevent these vulnerabilities is to simply use a language that does not allow for them. Languages that do not share these aspects are typically immune. Java, Python, and .NET, among other languages and platforms, don’t require special checks or changes to mitigate overflow vulnerabilities.  
Hardcoded Passwords can lead the program to be used by outside source. Hardcoded credentials are favored cyberattack targets for password guessing exploits, allowing hackers and malware to hijack firmware, devices (such as health monitoring equipment), systems, and software. Hardcoding not only presents a cyber risk for the specific device, firmware, application, etc. itself, but also to other components of the connected IT ecosystem. Additionally, innocent third-parties can be impacted by hardcoding negligence as they could be assailed by DDOS attacks from botnets of devices enslaved via a hardcoded credential compromise.

**Prevention:**

* **Bring application passwords under management:** Introduce a third-party hat uncovers default and hardcoded credentials across the enterprise, and forces applications, scripts, etc. to call (or request) the use of the password from a centralized password safe.
* **Refuse to buy software or hardware that includes hardcoded credentials:** Refusing to buy software and hardware with hardcoded password vulnerabilities helps put pressure on vendors to eliminate this poor security practice.
* **Perform Pen Testing:** For organizations with the most stringent of security environments, hiring outside pen testers to poke and prod for vulnerabilities, such as hardcoded and default credentials, provides an extra measure of proactive cyber defense.