Building Strong Programming Skills with Weak Encryption

(A Nifty Assignment)

Amy Larson
Computer Science Program
Department of Mathematics, Statistics, and Computer Science
AUGSBURG UNIVERSITY

WHAT IS THE PROBLEM?

Too many students
are not successfully completing
Introduction to Programming (Python)
at Augsburg
and many other institutions.

The average DFW rate nationally is 28% for introductory programming courses [*].

At Augsburg, it WAS 38% (in 2021).

[*] J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming: 12 years later," *ACM Inroads*, vol. 10, no. 2, pp. 30–36, Apr. 2019.

The Solution (or part of it, anyway – we think)

1. Overhauled assessment and adopted **equitable grading practices**,
2. Redesigned the curriculum to be **more engaging** and provide **more scaffolding**,
3. Increased academic support through **peer instructors**,
4. Fostered **community** through First-Year Experiences and peer tutoring.

Learning Objectives (Introduction to Programming)

It is later in the Semester ...

I am teaching about dictionaries and File I/O

Students will be able to:

1. define a new dictionary.
2. put and get elements into the dictionary.
3. read and parse a csv file.
4. write to a csv file.

But how I can I make this fun and engaging ...

Cybersecurity

Students are very interested and industry is interested (causal ??).

Learning Objectives (Introduction to Programming)

Students will be able to:

1. create and use **dictionaries** in Python.
2. (.csv) **file I/O.**
3. **number conversion** (decimal, binary, and hexadecimal)
4. convert between chars and **ASCII / UNICODE**.
5. describe the process of **hashing / encryption**.
6. list **vulnerabilities to hacking** of passwords.

Cybersecurity and Protecting Passwords

Individual Responsibility (as a consumer)

Corporate Responsibility

Resources:

- https://www.cisco.com/c/en/us/products/security/what-is-cybersecurity.html
- https://cloud.google.com/learn/what-is-encryption
- https://www.ssl2buy.com/wiki/difference-between-hashing-and-encryption
- https://www.techtarget.com/searchsecurity/definition/salt#:~:text=What%20is%20password%20salting%3F,stealing%20them%20from%20the%20database.
- https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/
- https://en.wikipedia.org/wiki/List_of_Unicode_characters
- https://ss64.com/ascii.html

A Simple Conversion (Caesar Cipher)

	M	**Y**	**P**	**A**	**S**	**S**	**W**	**O**	**R**	**D**	**!**
	77	89	80	41	83	83	87	79	82	68	33
+	2	5	1	0	3	2	5	2	2	3	4
	79	94	81	41	86	85	92	81	84	71	37
	O	**^**	**Q**	**A**	**V**	**U**	****	**Q**	**T**	**G**	**%**

KEY

What are other ways that computers interpret numbers or represent information?

Number Representation Practice (Worksheet)

Break decimal into components: 2,308 = 2*10^3 + 3*10^2 + 8*10^0 = 2000 + 300 + 8 71,030 = _____	Break binary into its components: 1011 = 2^3 + 2^1 + 2^0 = 8 + 2 + 1 = 11 10110 = _____
1 1 1 0 0 1 + 1 1 1 0 =========	Convert the binary into 2 hex digits, by first converting to decimal. 0110 1101 = _____

Hash Password – Base Level

1. **Salting the Password**
2. **Create a Bitstream**
3. **Convert a Bitstream to a Hexadecimal String**
4. **Convert a Hexadecimal String to an ASCII**

```
hex2deci = {
   '0':0,'1':1,'2':2,'3':3,'4':4,'5':5,'6':6,'7':7,'8':8,
   '9':9,'A':10,'B':11,'C':12,'D':13,'E':14,'F':15
   }
```

```
def create_bitstream(password,encodings):
   '''
   Convert each character of the password to its corresponding bitstream
   as defined in the encodings dictionary.
   For example, if password[0] = 'a' then add encodings['a'] to the bitstream.

   RETURN the bitstream (a string of 0s and 1s)
   '''
```

```
def hexify(bitstream):
   '''
   Convert the bitstream to a hex stream.
   Take every 4 bits and convert them to the corresponding hex value.
   For example, bitstream 001111110100 would be converted to '3F4'

   RETURN the hex string
   '''
```

Verify Password – Level Up (File I/O)

1. **Hash Password**
2. **Compare to Stored Password**
3. **New User Account**
4. **Write username + password to file of stored passwords.**

```
def write_users():
    '''
    Take the contents of the dictionary and write it to the file.
    Use 'w' when opening the file to overwrite the contents.
    Separate the username and encrypted password with a space character.
    '''

def add_account():
    '''
    Get a username from the user:
        Make sure it is 8 to 10 characters long.
        Make sure it is unique (i.e. not already in the dictionary)
        Continually ask until these criteria are met
    Get a password from the user:
        Make sure it is 6 to 15 characters long.
        Continually ask until these criteria are met
    Add the username and password to the dictionary.
    Write the dictionary to the file.
    '''
```

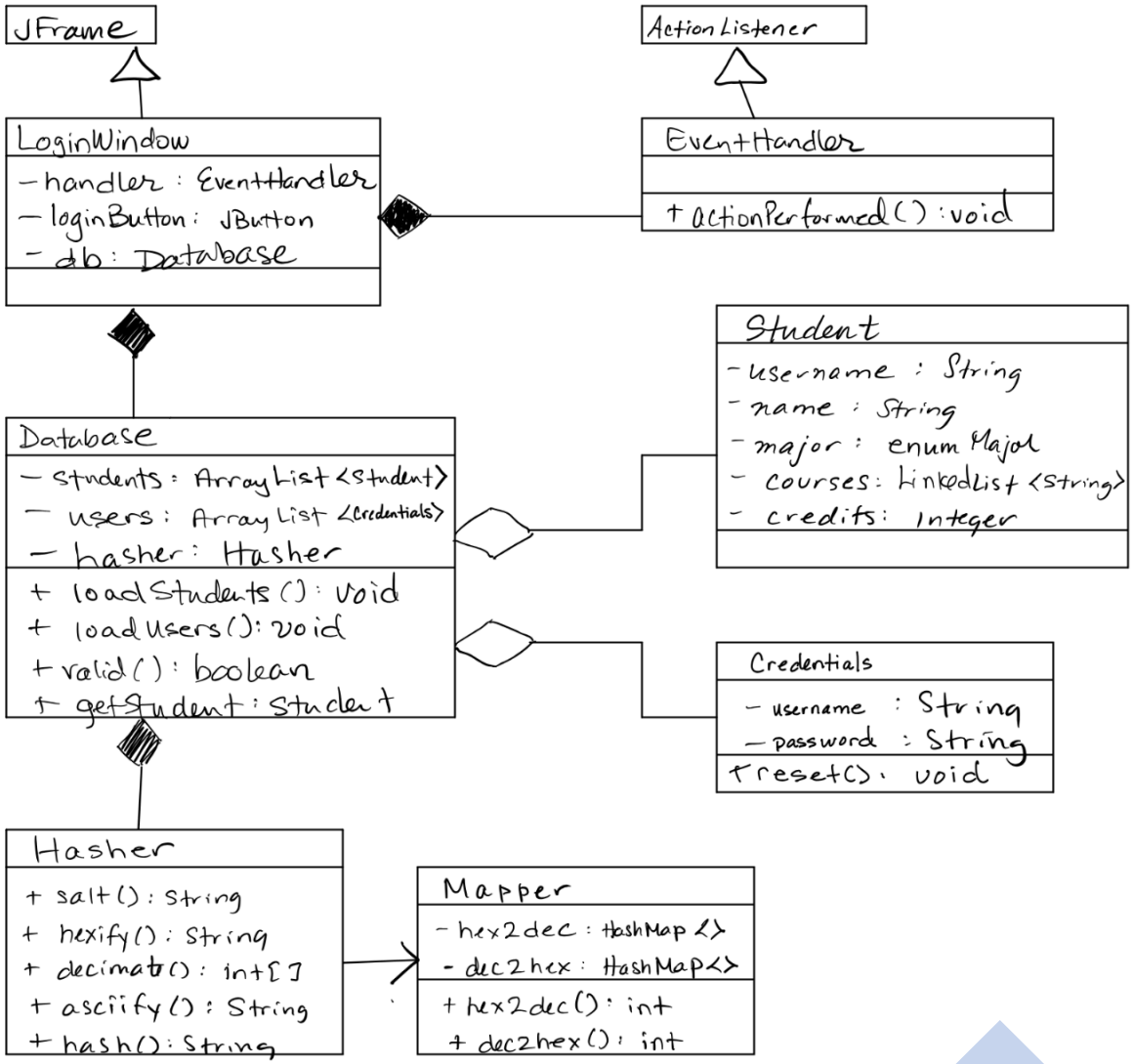
And Again ...

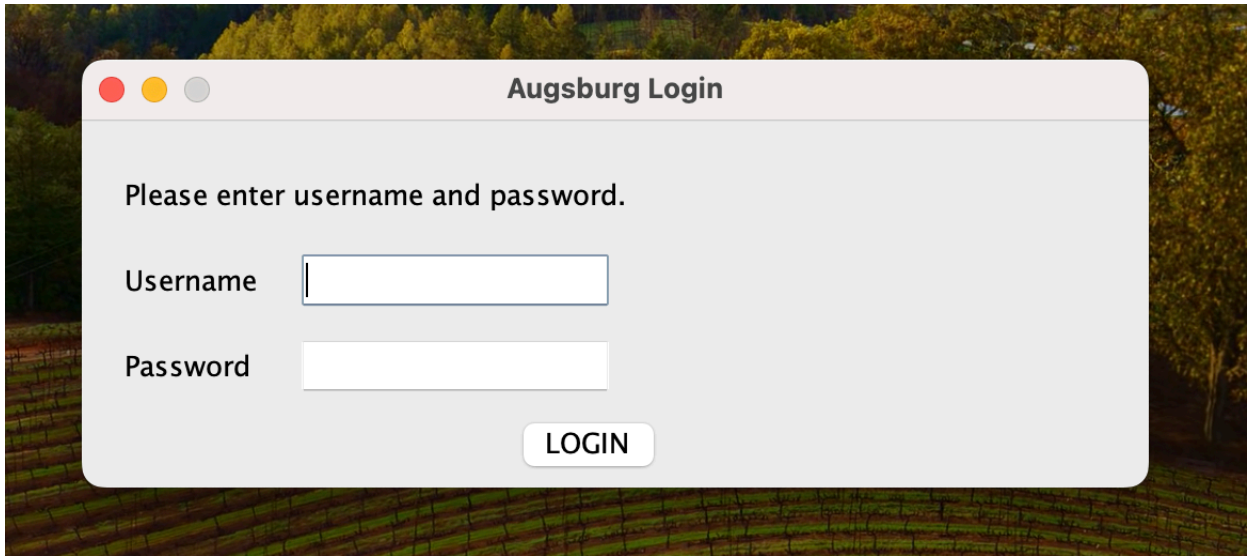
Data Structures

Learning Objectives (Data Structures)

Students will be able to:

1. (.csv) **file I/O.**
2. create **HashMap** and apply operations in Java.
3. create **ArrayList** and apply operations, including iteration.
4. create **LinkedList** and apply operations, including iteration.
5. modify text in **Graphics window**.
6. create an **event listener**.


JFrame
ActionListener
LoginWindow
- handler : EventHandler
- loginButton : JButton
- db : Database
EventHandler
+ actionPerformed() : void
Student
- username : String
- name : String
- major : enum Major
- courses : LinkedList <String>
- credits : Integer
Database
- students : ArrayList <Student>
- users : ArrayList <Credentials>
- hasher : Hasher
+ loadStudents() : void
+ loadUsers() : void
+ valid() : boolean
+ getStudent : Student
Credentials
- username : String
- password : String
+ reset() : void
Hasher
+ salt() : String
+ hexify() : String
+ decimate() : int[]
+ asciify() : String
+ hash() : String
Mapper
- hex2dec : HashMap <>
- dec2hex : HashMap<>
+ hex2dec() : int
+ dec2hex() : int


Augsburg Login
Please enter username and password.
Username
Password
LOGIN

```java
@Override
public void actionPerformed(ActionEvent ae) {

    // Retrieve entered text
    String username = userField.getText();
    String password = passwordField.getText();

    // validate credentials
    if (db.valid(username, password) == true){
       Student student = db.getStudent(username);
       message.setText("Hello " + student.name() + "!");
    }
    else{
       //invalid message
       message.setText("Username and/or password was invalid. Try
       again.");
    }
```

Event Handler in Pop-up

```java
// find the user and compare hashed passwords
for (int i = 0; i < users.size(); i++){

    if ( users.get(i).username().equals(username)) {
        if (hasher.hash(password).equals(users.get(i).password())) {
            return true;
        }
    }
}
```

Verifying Credentials

Part of Hashing Password

```java
public int[] decimate(String hexString) {
    // create an array to store corresponding values
    int[] key2dec = new int[hexString.length()];

    // map each hex to its decimal value using hash map in Converter
    for (int i=0; i<hexString.length(); i++) {
        key2dec[i] = Mapper.hex2dec(hexString.charAt(i));
```

Read .csv for User/Pwd

```java
String[] splitter = scan.nextLine().split(",");
Credentials newCred = new Credentials(splitter[0],splitter[1]);
users.add(newCred);
```

All on github

https://github.com/AugsburgCS/cybersecurity/