



# DATA STRUCTURES LESSON

Dan Bonarigo, Riverside Brookfield High School, Riverside, IL



This material is based upon work supported by  
the **National Science Foundation** under Grant No.1548315.

Additional materials may be found at [www.ncyte.net](http://www.ncyte.net)



# DATA STRUCTURES

## USERNAME PROGRAMMING ACTIVITY

Students will write programs in pseudocode to accomplish various tasks associated with username/password security and creation.

### PROGRAMMING TASK 1

Write a program that displays a list of 5 usernames. The program will then ask the user to create a password for each user that will be stored in a separate list. Once all the passwords are entered the program should display the list of passwords.

#### SAMPLE PROGRAM FOR TASK 1

Here is a sample output for the program (user entries are underlined):

*Enter a password for Bobby1: pa\$\$word*

*Enter a password for Jimmy: !3tme!n*

*Enter a password for Bonnie: pL3a\$e*

*Enter a password for Katie: He770*

*Enter a password for Carrie: Th@nk\$*

*Here is the list of passwords ["pa\$\$word", "!3tme!n", "pL3a\$e", "He770", "Th@nk\$"]*

### PROGRAMMING TASK 2

Write a program that displays a list of 5 usernames and a list of 5 passwords. The program will ask the user to type in a username and then the corresponding password. The program should verify the user has entered one of the usernames in the list, if not, then the user should be prompted to re-enter the username. Next, the user enters the password; if the user enters the password correctly, the program should display a message such as, "You have entered the correct password!" If the user enters an incorrect password, the program should display a



message such as, "You have entered an incorrect password! Try again." The program should allow the user to enter the passwords until they have entered it correctly.

---

## SAMPLE PROGRAM FOR TASK 2

Here is a sample output for the program (user entries are underlined):

*Here is a list of usernames ["Bobby1", "Jimmy", "Bonnie", "Katie", "Carrie"]*

*Here is the list of passwords ["pa\$\$word", "l3tme!n", "pL3a\$e", "He770", "Th@nk\$"]*

*Enter a username: Bonnie*

*That is not a valid username.*

*Enter a username: Bonnie*

*Enter the password for Bonnie: pa\$\$word*

*You have entered an incorrect password! Try again.*

*Enter the password for Bonnie: l3tme!n*

*You have entered an incorrect password! Try again.*

*Enter the password for Bonnie: pL3a\$e*

*You have entered the correct password!*

## PROGRAMMING TASK 3

Write a program that allows the user to enter as many unique usernames (that should be stored in a list) as they would like. Next, the user is prompted to enter a password for each username (stored in a separate list).

Once the lists are created the user is prompted to enter a username. If the user enters a username that is not in the list of usernames a message is displayed such as, "That username does not exist." The user is prompted to enter a username again. If the user enters a correct username they are prompted to enter the password. If the user enters the incorrect password a message is displayed such



as, "That is the incorrect password." The user is prompted to re-enter the correct password.

---

### SAMPLE PROGRAM FOR TASK 3

Here is a sample output for the program (user entries are underlined):

Enter a username: Frank

Enter another username? Yes

Enter a username: Sammy

Enter another username? Yes

Enter a username: Kimmy

Enter another username? No

Enter a password for Frank: Mi\$take

Enter a password for Sammy: \$ecur3

Enter a password for Kimmy: @nsw3r

Which account would you like to access? Kim

That username does not exist.

Which account would you like to access? Kimmy

Enter the password: Mi\$take

That is the incorrect password.

Enter the password: @nswer

That is the incorrect password.

Enter the password: @nsw3r

You now have access to your account!



## WHAT TO SUBMIT

Students should turn in their paper for grading and/or verification of completion. Teachers should use this as an opportunity for formative assessment to verify students' ability to write a program using lists and loops.

