



INPUT VALIDATION AND CONDITIONALS

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



INPUT VALIDATION AND CONDITIONALS

STRINGS AND VARIABLES PROGRAMMING ACTIVITY

Complete programming tasks one and two. Note: Students need internet connected devices to access the class' choice IDE.

PROGRAMMING TASK 1

Write a program that displays a password to the user such as "Pa\$sword123" or any other string. The program then asks the user to enter the password. The program should check whether the user has entered the given password correctly and displays a statement such as "Your password is correct!"; when the password is entered incorrectly, the program displays a statement such as "You have entered the password incorrectly!"

If the first attempt was incorrect, the user is prompted to enter the password again. The second attempt is checked and if correct a statement such as "Your second attempt is correct!" is displayed; if incorrect a statement such as "Two incorrect attempts! You are locked out of your account."

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

Your password is Pa\$sword123

Enter your password: Password123

You have entered the password incorrectly!

Enter your password: Pa\$sword123

Your second attempt is correct!



PROGRAMMING TASK 2

Write a program that asks a user to create a password that is at least 6 characters long and store it in a variable called password, if the user enters a password that is too short display a statement such as, "Your password is too short! Enter a longer password." Then ask the user to enter a password of the appropriate length. If they enter a password that is too short again the program should end and display a statement such as, "Too bad your password is too short."

If the password they entered is sufficient in length, the user is prompted to re-enter their password to validate their first entry. If the user enters the password correctly display a statement such as, "Your passwords match!" If the user re-enters the password incorrectly display a statement such as, "This password does not match your first entry." The user is prompted to re-enter the password again and the program checks if the re-entry matches the original password with the same display statements as above.

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

Enter a password that is at least 6 characters long: Pa\$sw

Your password is too short! Enter a longer password.

Enter a password of the appropriate length: Mypassword

Re-enter your password for validation: MyPassword

This password does not match your first entry.

Re-enter your password for validation: Mypassword

WHAT TO SUBMIT

Copy your program code into this worksheet and show text cases running to verify the program works as intended.

