Ian Sharff

COSC 010: Introduction to Computer Science

Project 1 Report

**I.** The conversational agent's purpose was to serve as a multifunctional PDA-like device that requires a username and password combination to be created and entered before the program can be functional. The bulk of this project's code serves to collect the username and password and verify that they were, indeed, what the user intended to submit. After "login" the program allows the user to choose a function to use. As of now there is only one, the "Calculator," that does the four basic operations as well as factorials. Expected inputs include: a username, a password, verifications for both, a desired "app" name, numbers for the Calculator function, and an operator for the Calculator function. In each step, expected outputs are various printed statements that instruct the user which values to enter.

The input function was used to collect data and store them in variables. The print function was used to print statements to the user. Conditional statements were used to verify whether user inputs were allowed under the "rules" of value entry as well as to branch the program depending on certain outputs. "While" loops were used to indefinitely allow for re-entry of incorrectly entered values. A "for" loop was used to cycle through characters in the password string value to see if there was a special character.

Limitations include the fact that the username is not stored in any location after the program is ended. Also, the Calculator function is rather clunky in its implementation. In general, all inputs are treated with conditional branching to cover all situations.

The program could definitely be expanded by simplifying the login section, having a username, password and password verification all happening in one step. Also, more functions can be added in addition to the Calculator which could be used while logged in.

# II.

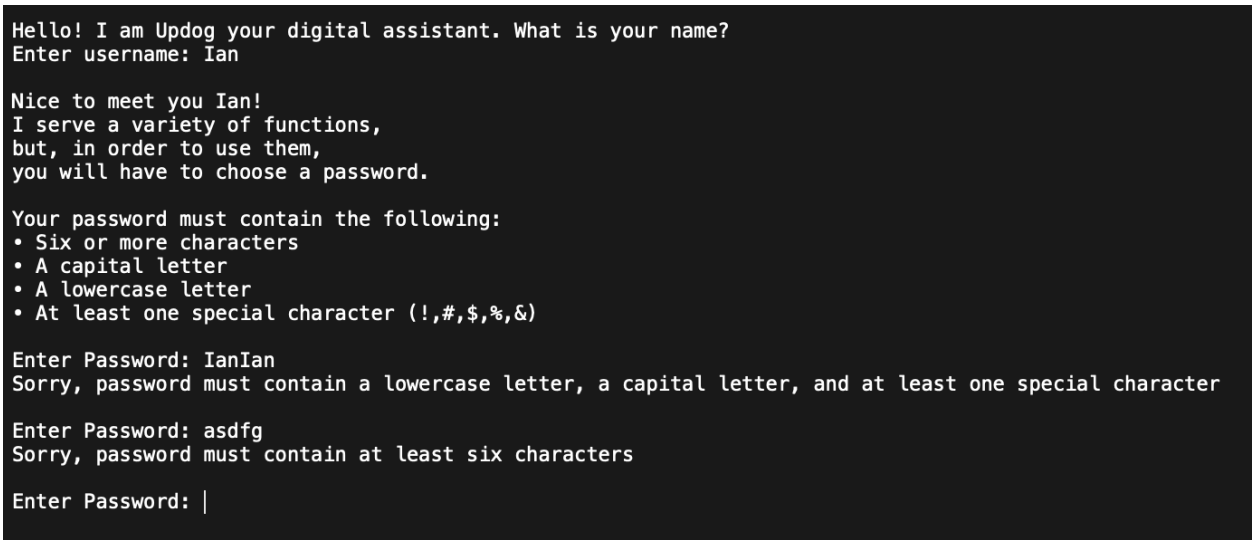


*Figure 1: Username entry with invalid entry messages due to password not having the proper characters*

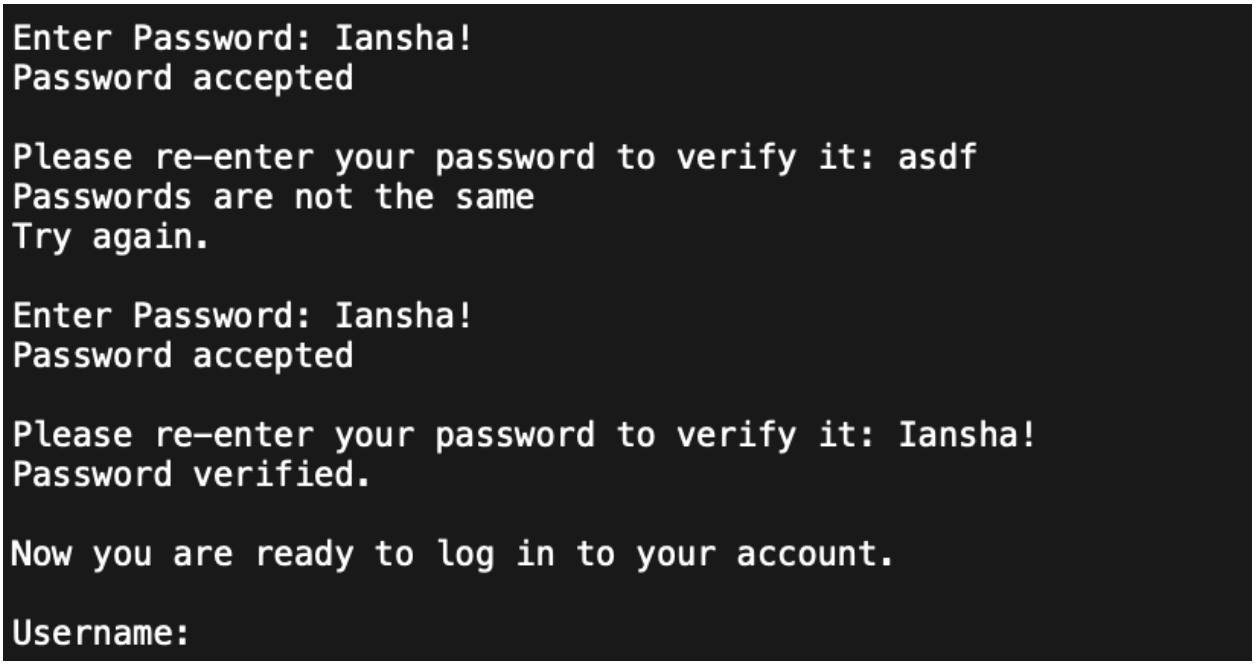


*Figure 2: Password entries must be the same, then user prompted again for username and password*

```
Username: Ian

Password: Iansha!
Welcome Ian!

Now, Ian, what would you like me to do?
In my current version, I only have one function.
Please choose from the following:
•Calculator

Choose function: Calculator
Please choose an operator (-,+,*,/,!)

Enter operator: +

Choose the first number: 3

Choose the second number: 4
3.0 + 4.0 is equal to 7.0
If you would like to do another calculation, please type 'Yes' without quotations
Otherwise, type anything else

-->Yes
Okay, proceed.
Please choose an operator (-,+,*,/,!)

Enter operator: !

Choose a number: 3
The factorial of 3.0 is 6
If you would like to do another calculation, please type 'Yes' without quotations
Otherwise, type anything else

-->no
See you later Ian!
```

*Figure 3: After login, Calculator function chosen, then addition performed, then factorial, then program ends*

# III.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Oct  6
Ian Sharff
COSC-010
Project 1
10/09/2020
"""
import sys
import time
import math
delaytime = 1


#Greet function collects and returns the selected username
#Sleep delays to simulate thinking
def Greet():
    username = input("Hello! I am Updog your digital assistant. What is your name?\nEnter username: ")
    time.sleep(2 * delaytime)
    print("")
    print(f"Nice to meet you {username}!")
    time.sleep(delaytime)
    print("I serve a variety of functions,")
    time.sleep(delaytime)
    print("but, in order to use them,")
    time.sleep(delaytime)
    print("you will have to choose a password.")
    time.sleep(2 * delaytime)
    print("")
    return username

#Password function collects password from the user
def EnterPassword():
    print("Your password must contain the following:")
    time.sleep(delaytime)
    #lists requirements for password
    print("• Six or more characters\n• A capital letter\n• A lowercase letter\n• At least one special character (!,#,$,%,&)")
    time.sleep(delaytime)

    specialcharacters = ["!", "#", "$", "%", "&"]       #one of these characters needed for password
    password_valid = False                          #flag variables for while loop
```

```
    password_verified = False

    while password_valid == False or password_verified == False :

        password = input("Enter Password: ")
        time.sleep(delaytime)
        over_six = len(password) >= 6   #checks if over 6 characters
        all_upper = password.isupper()  #if True, then needs a capital letter
        all_lower = password.islower()  #if False, then needs a lowercase letter

        for char in password :                #for loop checks each character
            if char in specialcharacters :      #to see if in list of specials
                has_special = True
                break
            else :
                has_special = False

        if over_six == False :      #if too short, prints a message
            print("Sorry, password must contain at least six characters")

        #checks for other conditions at the same time
        elif all_upper == True or all_lower == True or has_special == False :
            print("Sorry, password must contain a lowercase letter, a capital letter, and at least one
special character")
        #if conditions are all met, then the following conjunction must be true
        #password_valid could be assigned to True
        else :
            password_valid = over_six and has_special and not all_upper and not all_lower
            print("Password accepted")
            time.sleep(delaytime)
            #password verification
            verified_pw = input("Please re-enter your password to verify it: ")
            time.sleep(delaytime)
            #if password is not the same then loop continues
            if verified_pw != password :
                print("Passwords are not the same")
                time.sleep(delaytime)
                print("Try again.")
                time.sleep(delaytime)
            #otherwise it must be the same, then loop ends
            else :
                print("Password verified.")
                print("")
                time.sleep(delaytime)
                password_verified = True
    return password
```

```
#reenter username and password, username and password variables must be args
def Login(username, password):
    print("Now you are ready to log in to your account.")
    time.sleep(delaytime)
    user_entry = ""            #Defining variables for user entry
    password_entry = ""
    attempts_left = 3          #After three attempts, program stops

    #while entered username and entered password are not BOTH correct, loop continues
    #however, if the user is out of attempts, then loop will end and program will end
    while not (user_entry == username and password_entry == password) and attempts_left > 0 :
        user_entry = input("Username: ")
        time.sleep(delaytime)
        password_entry = input("Password: ")
        #if password and username aren't correct, then minus 1 attempt
        if user_entry != username or password_entry != password :
            print("Username/Password combiniation is incorrect")
            time.sleep(delaytime)
            attempts_left -= 1

            print(f"You have {attempts_left} attempts left.")
            time.sleep(delaytime)
            #if no attempts left, program ends
            if attempts_left == 0 :
                print("You are out of attempts and have been locked out.")
                sys.exit(0)
        #not(not(A) or not(B)) is equivalent to (A and B) if first conditional
        #is not met then password and username must be correct
        else :
            print(f"Welcome {username}!")
            print("")
            time.sleep(delaytime)
# Instructs user to choose the function they want the device to perform
def FunctionSelection(username):
    print(f"Now, {username}, what would you like me to do?")
    time.sleep(delaytime)
    print("In my current version, I only have one function.")
    time.sleep(delaytime)
    print("Please choose from the following: \n•Calculator")
    time.sleep(delaytime)

    function_choice = input("Choose function: ")
    time.sleep(delaytime)
    # collects user input and returns string value
    return function_choice
```

```
#Calculator function, user inputs one operator, then two numbers for normal
#operators and one number for factorial

def Calculator():
    normal_operators = ["-", "+", "/", "*"]
    exitloop = False                #Flag variable to end loop
    while exitloop == False :
        print("Please choose an operator (-,+,*,/,!)")
        operator = input("Enter operator: ")

        #checks inputted operator if in list of normal operators
        if operator in normal_operators :
            # handles error from non-numeric entry and ends program
            try :
                firstnumber = float(input("Choose the first number: "))
                secondnumber = float(input("Choose the second number: "))
                time.sleep(delaytime)
            except:
                print("Values unacceptable. Must be numeric. Cannot compute.")
                sys.exit(0)

            if operator == "+" :                    #depending on operator,
                answer = firstnumber + secondnumber     #conditional executes
            elif operator == "-" :                  #an operation
                answer = firstnumber - secondnumber
            elif operator == "*" :
                answer = firstnumber * secondnumber
            else :
                #handles error from division by zero
                try :
                    answer = firstnumber / secondnumber
                except :
                    print("Error. Cannot divide by zero. Goodbye.")
                    sys.exit(0)
            #prints result
            print(f"{firstnumber} {operator} {secondnumber} is equal to {answer}")
        # if not normal and it's a factorial, then only requires one input number
        elif operator == "!" :
            #handles error from non-numeric entry
            try :
                number = float(input("Choose a number: "))
            except:
                print("Invalid entry, must be numeric. Cannot compute")
                sys.exit(0)
            #handles error from negative integers or float values
```

```
            try :
                factorial = math.factorial(number)
            except :
                print("Value unacceptable ")

            print(f"The factorial of {number} is {factorial}")

        #if operator is not one of the five allowed, then program ends
        else :
            print("Invalid operator. Goodbye.")
            sys.exit(0)
        #prompts user if they would like to do another calculation
        print("If you would like to do another calculation, please type 'Yes' without quotations")
        print("Otherwise, type anything else")
        repeat = input("-->")

        #if user inputs Yes then repeats. if not then program continues
        if repeat == "Yes" :
            print("Okay, proceed.")
            time.sleep(delaytime)
        else :
            exitloop = True
#says goodbye to user
def SignOff(username):
    print(f"See you later {username}!")

def main():
    username = Greet()              #greet function collects username
    password = EnterPassword()      #collects password
    Login(username, password)       #makes re-enter, if wrong 3 times then program ends
    function_choice = FunctionSelection(username) #prompts user for function choice
    if function_choice == "Calculator": #if Calculator not entered then goes to SignOff
        Calculator()
    else :
        pass
    SignOff(username)

main()
```