week3_programs

January 15, 2025

[2]:
```
"""Modify your greeting program so that if the user does not enter a name (i.e.␣
 ↪they
just press enter), the program responds "Hello, Stranger!". Otherwise it should␣
 ↪print
a greeting with their name as before."""
def display():
    name=user_input()
    if name=="":
        print("Hello, Stranger!.")
    else:
        print(f"Hello, {name}!")
def user_input():
    name=input("Enter your name:")
    return name
display()
```

```
Enter your name:
```

```
Hello, Stranger!.
```

[3]:
```
"""Write a program that simulates the way in which a user might choose a␣
 ↪password.
The program should prompt for a new password, and then prompt again. If the two
passwords entered are the same the program should say "Password Set" or
similar, otherwise it should report an error."""
def display():
    while True:
        password1,password2=password()
        if password1==password2:
            print("Password Set")
            break
        else:
            print("Password didn't match! Please, re-enter the password.")
def password():
    password1=input("Enter your password:")
    password2=input("Confirm your password:")
    return password1, password2
display()
```

```
Enter your password: programming
Confirm your password: programming

Password Set
```

[4]:
```
"""Modify your previous program so that the password must be between 8 and 12
characters (inclusive) long."""

def password():
    while True:
        password1, password2 = user_input()

        if password1 == password2 and 8 <= len(password1) <= 12:
            print("Password Set")
            break
        else:
            print("There is an error while inputting password. You have to␣
 ↪re-enter the password.")
def user_input():
    password1 = input("Enter your password (8 to 12 characters): ")
    password2 = input("Confirm your password: ")
    return password1, password2
password()
```

```
Enter your password (8 to 12 characters):  python programming
Confirm your password:  python programming

There is an error while inputting password. You have to re-enter the password.

Enter your password (8 to 12 characters):  programm
Confirm your password:  programm

Password Set
```

[5]:
```
"""Modify your program again so that the chosen password cannot be one of a␣
 ↪list of
common passwords, defined thus:
BAD_PASSWORDS = ['password', 'letmein', 'sesame', 'hello', 'justinbieber']"""
def display():
    while True:
        password1,password2=password()
        BAD_PASSWORDS = ['password', 'letmein', 'sesame', 'hello',␣
 ↪'justinbieber']
        if password1!=password2:
            print("Passwords do not match. Please re-enter.")
        elif password1 in BAD_PASSWORDS:
            print("Weak Password. Please choose a stronger one.")
        elif " " in password1:
            print("Password cannot contain spaces. Please try again.")
        elif 8 > len(password1) > 12 :
```

```
            print("Password length must be between 8 and 12 characters.")
        else:
            print("Password set successfully!")
            break
def password():
    password1 = input("Enter your password (8 to 12 characters): ").strip()
    password2 = input("Confirm your password: ").strip()
    return password1, password2
display()
```

```
Enter your password (8 to 12 characters):  hello
Confirm your password:  hello

Weak Password. Please choose a stronger one.

Enter your password (8 to 12 characters):  pass word
Confirm your password:  pass word

Password cannot contain spaces. Please try again.

Enter your password (8 to 12 characters):  programm
Confirm your password:  programm

Password set successfully!
```

[6]:
```
"""Modify your program a final time so that it executes until the user␣
 ↪successfully
chooses a password. That is, if the password chosen fails any of the checks, the
program should return to asking for the password the first time."""
def display():
    BAD_PASSWORDS = ['password', 'letmein', 'sesame', 'hello', 'justinbieber']
    print("Welcome! Please set a secure password.")

    while True:
        password1, password2 = password()
        if not password1 or not password2:
            print("Passwords cannot be empty. Please try again.")
        elif 8 > len(password1) > 12:
            print("Password length must be between 8 and 12 characters. Please␣
 ↪try again.")
        elif " " in password1:
            print("Password cannot contain spaces. Please try again.")
        elif password1 != password2:
            print("Passwords do not match. Please re-enter.")
        elif password1 in BAD_PASSWORDS:
            print("Weak password. Please choose a stronger one.")
        else:
            print("Password set successfully!")
            break
```

```
def password():
    password1 = input("Enter your password (8 to 12 characters): ").strip()
    password2 = input("Confirm your password: ").strip()
    return password1, password2
display()
```

```
Welcome! Please set a secure password.

Enter your password (8 to 12 characters):  password
Confirm your password:  password

Weak password. Please choose a stronger one.

Enter your password (8 to 12 characters):  program
Confirm your password:  program

Password set successfully!
```

[7]:
```
"""Write a program that displays the "Seven Times Table". That is, the result of
multiplying 7 by every number from 0 to 12 inclusive. The output might start:
0 x 7 = 0
1 x 7 = 7
2 x 7 = 14
and so on."""

def table():
    num=7
    for i in range(0,13):
        print(f"{i} * {num} = {i*num}")
table()
```

```
0 * 7 = 0
1 * 7 = 7
2 * 7 = 14
3 * 7 = 21
4 * 7 = 28
5 * 7 = 35
6 * 7 = 42
7 * 7 = 49
8 * 7 = 56
9 * 7 = 63
10 * 7 = 70
11 * 7 = 77
12 * 7 = 84
```

[8]:
```
"""Modify your "Times Table" program so that the user enters the number of the␣
 ↪table
they require. This number should be between 0 and 12 inclusive."""
def table():
    num1=user_input()
```

```
    for i in range(0,13):
        print(f"{i} * {num1} = {i*num1}")
def user_input():
    num=int(input("Enter the number of which you want to calculate the␣
 ↪multiplication table:"))
    return num
table()
```

```
Enter the number of which you want to calculate the multiplication table: 3
```

```
0 * 3 = 0
1 * 3 = 3
2 * 3 = 6
3 * 3 = 9
4 * 3 = 12
5 * 3 = 15
6 * 3 = 18
7 * 3 = 21
8 * 3 = 24
9 * 3 = 27
10 * 3 = 30
11 * 3 = 33
12 * 3 = 36
```

[9]:
```
"""Modify the "Times Table" again so that the user still enters the number of␣
 ↪the table,
but if this number is negative the table is printed backwards. So entering "-7"
would produce the Seven Times Table starting at "12 times" down to "0 times"."""
def TimesTable():
    num=user_input()
    if num>0:
        for i in range(13):
            print(f"{i} * {num} = {i*num}")
    else:
        for i in range(-13,0):
            print(f"{-i} * {num} = {-i*num}")
def user_input():
    num=int(input("Enter the number of which you want to calculate the␣
 ↪multiplication table:"))
    return num
TimesTable()
```

```
Enter the number of which you want to calculate the multiplication table: -13
```

```
13 * -13 = -169
12 * -13 = -156
11 * -13 = -143
10 * -13 = -130
9 * -13 = -117
```

```
8 * -13 = -104
7 * -13 = -91
6 * -13 = -78
5 * -13 = -65
4 * -13 = -52
3 * -13 = -39
2 * -13 = -26
1 * -13 = -13
```

[ ]: