

Name: Sai Yashwanth Reddy Kontham
Student ID: 700744498

NEURAL NETWORKS AND DEEP LEARNING
ASSIGNMENT -1

1. Write a python program for the following:

– Input the string “Python” as a list of characters from console, delete at least 2 characters, reverse the resultant string and print it.

Sample input:
python

Sample output:
ntyp

– Take two numbers from user and perform at least 4 arithmetic operations on them.

Code:

```
# Input the string as a list of characters
import random;
input_string = list(input("Enter the string: "))

# Delete at least 2 characters
if len(input_string) >= 2:
    rand1=random.randint(0,len(input_string)-1)
    input_string.pop(rand1)
    rand2=random.randint(0,len(input_string)-1)
    input_string.pop(rand2)

# Reverse the resulting string
result_string = input_string[::-1]

# Print the reversed string
print("Resulting String:", "".join(result_string))

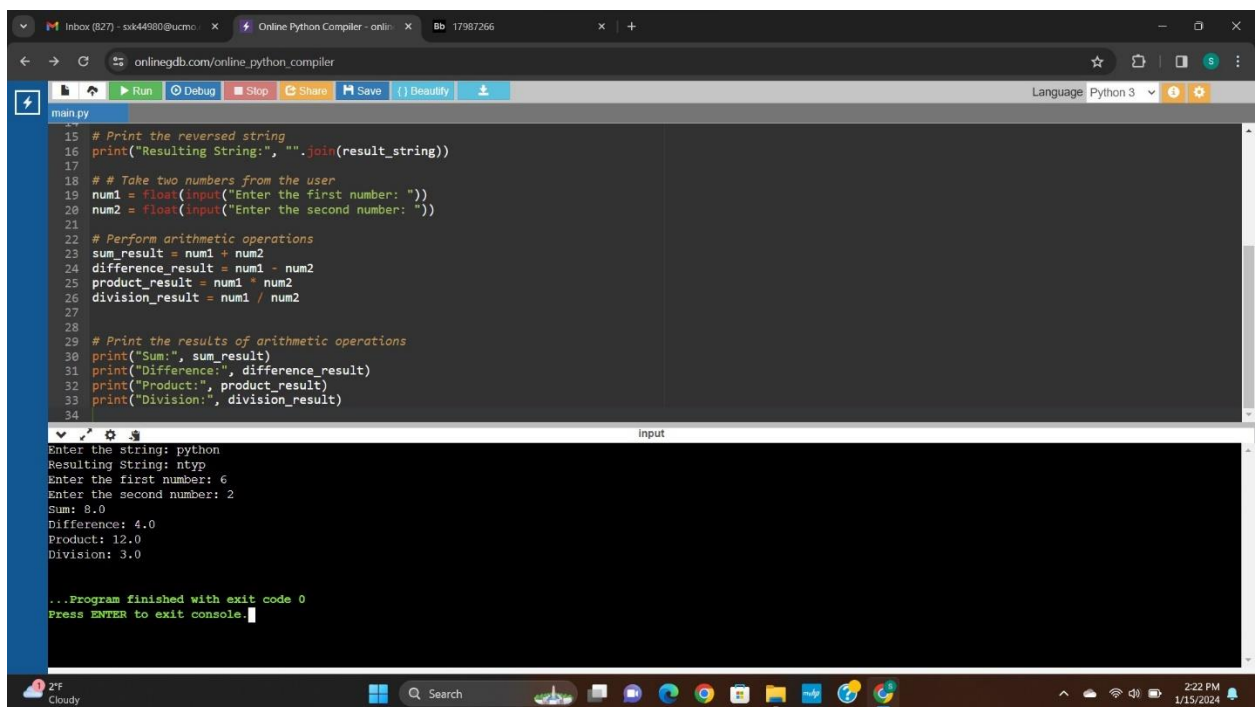
# # Take two numbers from the user
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

# Perform arithmetic operations
sum_result = num1 + num2
```

```
difference_result = num1 - num2
product_result = num1 * num2
division_result = num1 / num2
```

```
# Print the results of arithmetic operations
print("Sum:", sum_result)
print("Difference:", difference_result)
print("Product:", product_result)
print("Division:", division_result)
```

Output:

A screenshot of a web browser displaying an online Python compiler. The browser has three tabs: 'Inbox (827) - svk44980@ucmo', 'Online Python Compiler - online', and 'Bb 17387266'. The address bar shows 'onlinegdb.com/online_python_compiler'. The code editor contains a Python script with comments and arithmetic operations. The output console shows the results of running the code, including user input and calculated values. The taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system clock indicates 2:22 PM on 1/15/2024.

```
main.py
15 # Print the reversed string
16 print("Resulting String:", "".join(result_string))
17
18 # # Take two numbers from the user
19 num1 = float(input("Enter the first number: "))
20 num2 = float(input("Enter the second number: "))
21
22 # Perform arithmetic operations
23 sum_result = num1 + num2
24 difference_result = num1 - num2
25 product_result = num1 * num2
26 division_result = num1 / num2
27
28
29 # Print the results of arithmetic operations
30 print("Sum:", sum_result)
31 print("Difference:", difference_result)
32 print("Product:", product_result)
33 print("Division:", division_result)
34
```

```
Enter the string: python
Resulting String: ntyp
Enter the first number: 6
Enter the second number: 2
Sum: 8.0
Difference: 4.0
Product: 12.0
Division: 3.0

...Program finished with exit code 0
Press ENTER to exit console.
```

2. Write a program that accepts a sentence and replace each occurrence of 'python' with 'pythons'.

Sample input:

I love playing with python

Sample output:

I love playing with pythons

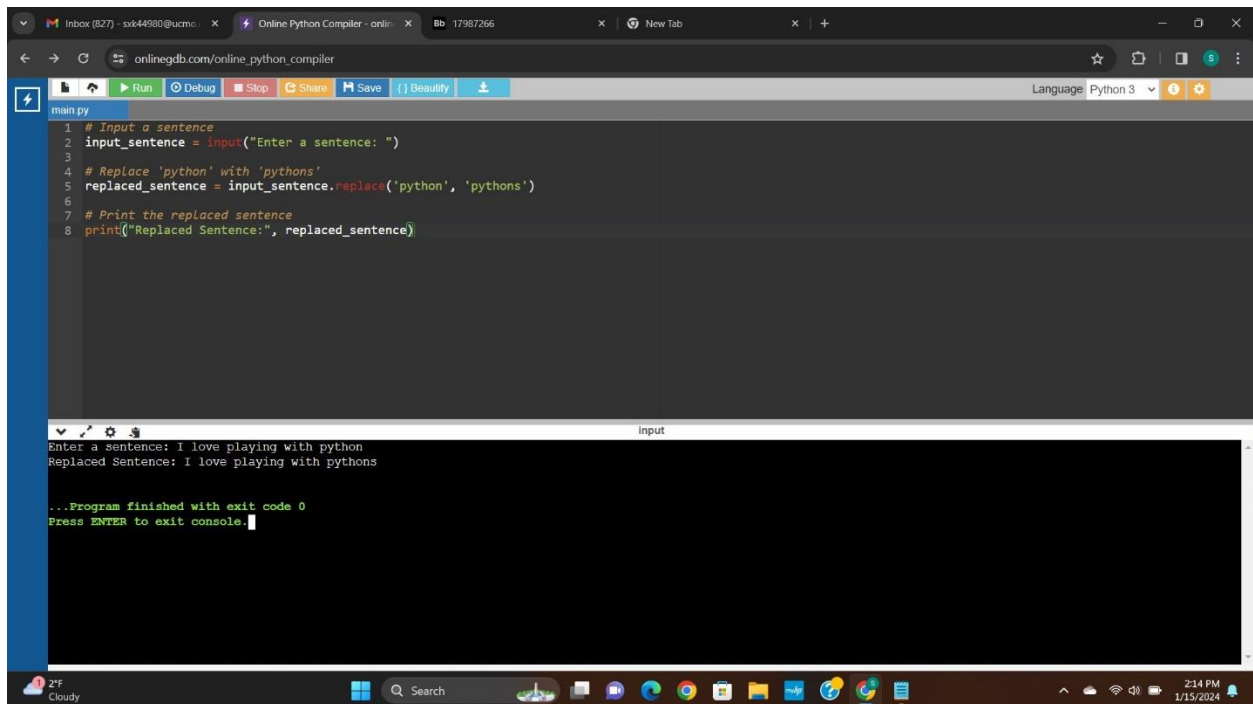
Code:

```
# Input a sentence
input_sentence = input("Enter a sentence: ")

# Replace 'python' with 'pythons'
replaced_sentence = input_sentence.replace('python', 'pythons')

# Print the replaced sentence
print("Replaced Sentence:", replaced_sentence)
```

Output:



The screenshot shows a web browser window with the URL `onlinegdb.com/online_python_compiler`. The browser has several tabs open, including 'Inbox (827) - ssk44980@ucmo', 'Online Python Compiler - onli...', 'Bb 17987266', and 'New Tab'. The compiler interface includes a toolbar with buttons for 'Run', 'Debug', 'Stop', 'Share', 'Save', and 'Beautify'. The code editor displays the following Python code:

```
1 # Input a sentence
2 input_sentence = input("Enter a sentence: ")
3
4 # Replace 'python' with 'pythons'
5 replaced_sentence = input_sentence.replace('python', 'pythons')
6
7 # Print the replaced sentence
8 print("Replaced Sentence:", replaced_sentence)
```

Below the code editor, the 'input' section shows the program's execution. It displays the prompt 'Enter a sentence: I love playing with python' and the resulting output 'Replaced Sentence: I love playing with pythons'. At the bottom, it states '...Program finished with exit code 0' and 'Press ENTER to exit console.' The Windows taskbar at the bottom shows the system clock as 2:14 PM on 1/15/2024, with a weather icon indicating 2°F and Cloudy.

The screenshot shows a web browser window with the URL `onlinegdb.com/online_python_compiler`. The browser has two tabs: "Inbox (827) - ssk44980@ucmo" and "Online Python Compiler - online". The compiler interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to Python 3. The code editor contains the following Python code:

```
1 # Input a sentence
2 input_sentence = input("Enter a sentence: ")
3
4 # Replace 'python' with 'pythons'
5 replaced_sentence = input_sentence.replace('python', 'pythons')
6
7 # Print the replaced sentence
8 print("Replaced Sentence:", replaced_sentence)
```

The output console shows the program's execution:

```
Enter a sentence: python is an interpreted language
Replaced Sentence: pythons is an interpreted language

...Program finished with exit code 0
Press ENTER to exit console.
```

The Windows taskbar at the bottom shows the system clock as 2:13 PM on 1/15/2024, with a weather widget indicating 2°F and Cloudy.

This screenshot shows the same online Python compiler interface as the first image, but with different input and output. The code in the editor is identical:

```
1 # Input a sentence
2 input_sentence = input("Enter a sentence: ")
3
4 # Replace 'python' with 'pythons'
5 replaced_sentence = input_sentence.replace('python', 'pythons')
6
7 # Print the replaced sentence
8 print("Replaced Sentence:", replaced_sentence)
```

The output console now shows:

```
Enter a sentence: I love python
Replaced Sentence: I love pythons

...Program finished with exit code 0
Press ENTER to exit console.
```

The Windows taskbar at the bottom remains the same, showing 2:13 PM on 1/15/2024.

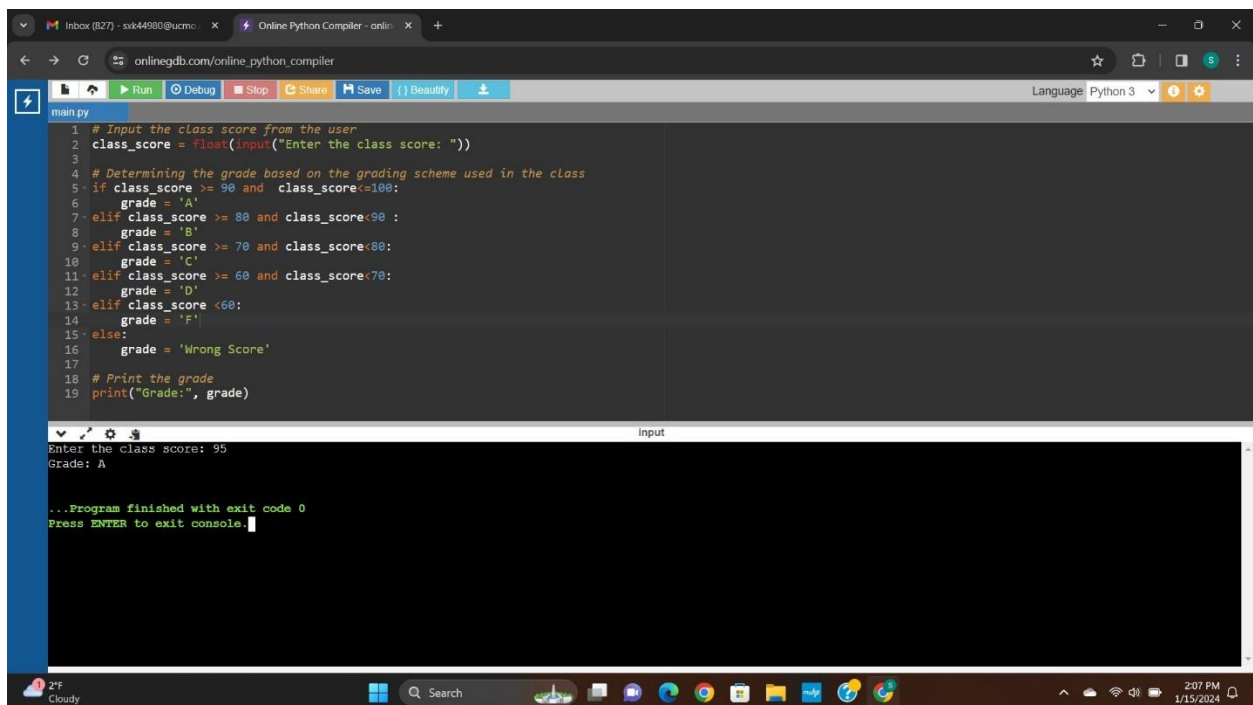
3. Use the if statement conditions to write a program to print the letter grade based on an input class score. Use the grading scheme we are using in this class.

Code:

```
# Input the class score from the user
class_score = float(input("Enter the class score: "))

# Determining the grade based on the grading scheme used in the class
if class_score >= 90 and class_score <= 100:
    grade = 'A'
elif class_score >= 80 and class_score < 90:
    grade = 'B'
elif class_score >= 70 and class_score < 80:
    grade = 'C'
elif class_score >= 60 and class_score < 70:
    grade = 'D'
elif class_score < 60:
    grade = 'F'
else:
    grade = 'Wrong Score'
# Print the grade
print("Grade:", grade)
```

Output:



The screenshot shows a web browser window with the URL `onlinegdb.com/online_python_compiler`. The browser's address bar and tabs are visible at the top. Below the browser window, there is a code editor with a dark background. The code in the editor is the same Python program as shown in the 'Code' block. The code is numbered from 1 to 19. Below the code editor, there is a console window with a light background. The console shows the input '95' and the output 'Grade: A'. At the bottom of the console, it says '...Program finished with exit code 0' and 'Press ENTER to exit console.' The bottom of the image shows a Windows taskbar with various icons and the system clock showing '2:07 PM 1/15/2024'.

```
1 # Input the class score from the user
2 class_score = float(input("Enter the class score: "))
3
4 # Determining the grade based on the grading scheme used in the class
5 if class_score >= 90 and class_score <= 100:
6     grade = 'A'
7 elif class_score >= 80 and class_score < 90:
8     grade = 'B'
9 elif class_score >= 70 and class_score < 80:
10    grade = 'C'
11 elif class_score >= 60 and class_score < 70:
12    grade = 'D'
13 elif class_score < 60:
14    grade = 'F'
15 else:
16    grade = 'Wrong Score'
17
18 # Print the grade
19 print("Grade:", grade)
```

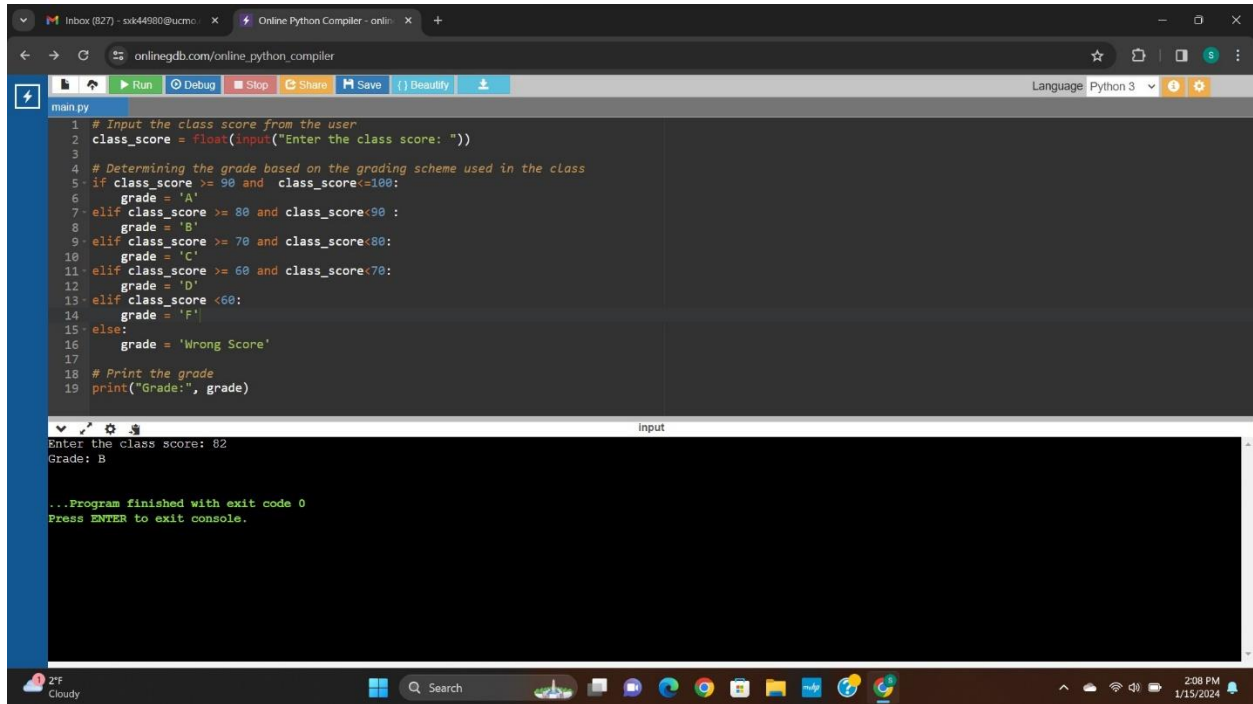
input

Enter the class score: 95

Grade: A

...Program finished with exit code 0

Press ENTER to exit console.



The screenshot shows a web browser window with the URL `onlinegdb.com/online_python_compiler`. The page has a dark theme. At the top, there are tabs for "Inbox (827)" and "Online Python Compiler - online". Below the browser window, there is a toolbar with buttons for "Run", "Debug", "Stop", "Share", "Save", and "Beautify". The main area is divided into two panels. The left panel, titled "main.py", contains the following Python code:

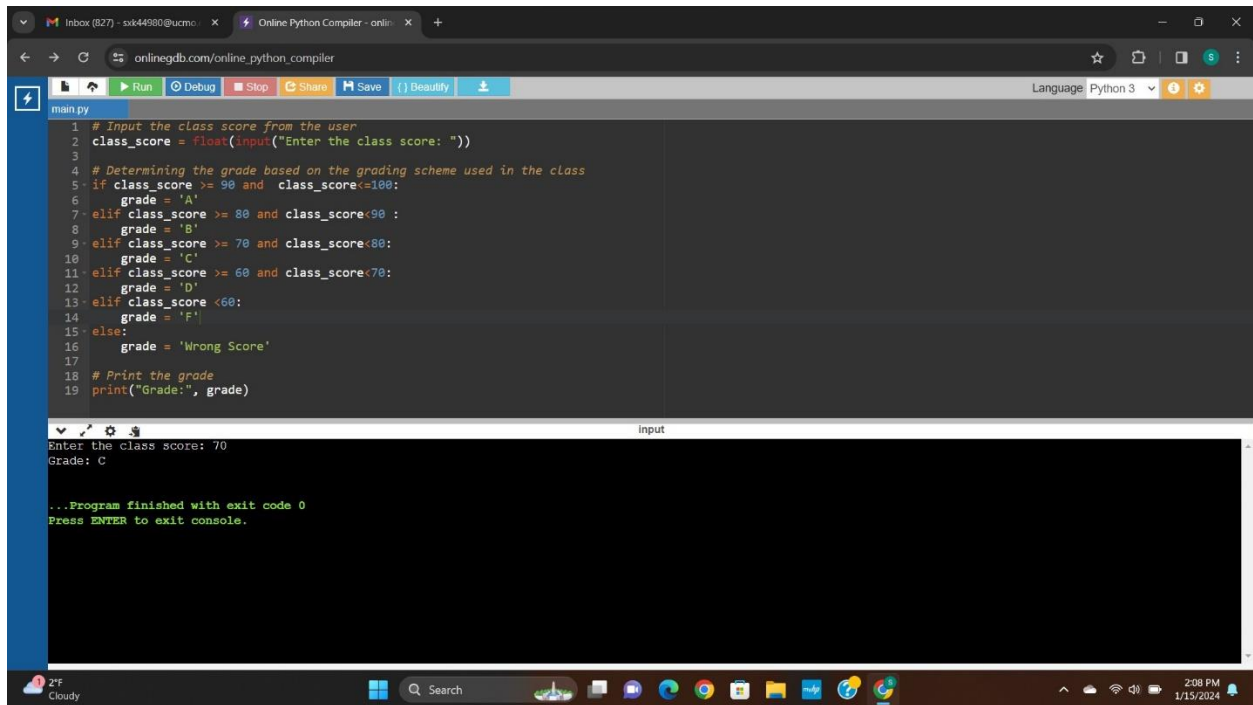
```
1 # Input the class score from the user
2 class_score = float(input("Enter the class score: "))
3
4 # Determining the grade based on the grading scheme used in the class
5 if class_score >= 90 and class_score <= 100:
6     grade = 'A'
7 elif class_score >= 80 and class_score < 90:
8     grade = 'B'
9 elif class_score >= 70 and class_score < 80:
10    grade = 'C'
11 elif class_score >= 60 and class_score < 70:
12    grade = 'D'
13 elif class_score < 60:
14    grade = 'F'
15 else:
16    grade = 'Wrong Score'
17
18 # Print the grade
19 print("Grade:", grade)
```

The right panel, titled "Input", shows the execution output:

```
Enter the class score: 82
Grade: B

...Program finished with exit code 0
Press ENTER to exit console.
```

At the bottom of the browser window, there is a Windows taskbar showing the time as 2:08 PM on 1/15/2024.



This screenshot shows the same online Python compiler interface as the first screenshot. The code in the left panel is identical. The right panel, titled "Input", shows the execution output for a different input:

```
Enter the class score: 70
Grade: C

...Program finished with exit code 0
Press ENTER to exit console.
```

The Windows taskbar at the bottom shows the same time and date as the first screenshot.

The screenshot shows a web browser window with the URL `onlinegdb.com/online_python_compiler`. The page has a dark theme. At the top, there are tabs for "Inbox (827)" and "Online Python Compiler - online". Below the browser window, there is a toolbar with buttons for "Run", "Debug", "Stop", "Share", "Save", and "Beautify". The main area is divided into two panels. The left panel, titled "main.py", contains the following Python code:

```
1 # Input the class score from the user
2 class_score = float(input("Enter the class score: "))
3
4 # Determining the grade based on the grading scheme used in the class
5 if class_score >= 90 and class_score <= 100:
6     grade = 'A'
7 elif class_score >= 80 and class_score < 90:
8     grade = 'B'
9 elif class_score >= 70 and class_score < 80:
10    grade = 'C'
11 elif class_score >= 60 and class_score < 70:
12    grade = 'D'
13 elif class_score < 60:
14    grade = 'F'
15 else:
16    grade = 'Wrong Score'
17
18 # Print the grade
19 print("Grade:", grade)
```

The right panel, titled "Input", shows the execution output:

```
Enter the class score: 65
Grade: D

...Program finished with exit code 0
Press ENTER to exit console.
```

At the bottom of the browser window, there is a Windows taskbar showing the time as 2:09 PM on 1/15/2024.

The screenshot shows the same online Python compiler interface as the first image. The code in the left panel is identical. The right panel, titled "Input", shows the execution output for a different input:

```
Enter the class score: 55
Grade: F

...Program finished with exit code 0
Press ENTER to exit console.
```

The Windows taskbar at the bottom shows the same time and date as the first image.

The screenshot shows a web browser window with the URL `onlinegdb.com/online_python_compiler`. The browser has two tabs: "Inbox (827) - svk44380@ucmc..." and "Online Python Compiler - online...". The compiler interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The code editor shows a Python script for determining a grade based on a class score. The script uses a series of if-elif-else statements to assign grades from 'A' to 'F' based on score ranges. The input field shows "Enter the class score: 110" and the output field shows "Grade: Wrong Score". The console at the bottom indicates the program finished with exit code 0.

```
1 # Input the class score from the user
2 class_score = float(input("Enter the class score: "))
3
4 # Determining the grade based on the grading scheme used in the class
5 if class_score >= 90 and class_score <= 100:
6     grade = 'A'
7 elif class_score >= 80 and class_score < 90:
8     grade = 'B'
9 elif class_score >= 70 and class_score < 80:
10    grade = 'C'
11 elif class_score >= 60 and class_score < 70:
12    grade = 'D'
13 elif class_score < 60:
14    grade = 'F'
15 else:
16    grade = 'Wrong Score'
17
18 # Print the grade
19 print("Grade:", grade)
```

Enter the class score: 110
Grade: Wrong Score

...Program finished with exit code 0
Press ENTER to exit console.

GithubLink:

https://github.com/konthamsaiyashwanthreddy/Neural_AT1

Video Link:

https://github.com/konthamsaiyashwanthreddy/Neural_AT1/blob/main/video_assgn1.mp4