**Objective: To identify and fix errors in a Python program that manipulates strings**

Code 1:

def reverse\_string(s):

reversed = ""

for i in range(len(s)-1,-1,-1):

reversed += s[i]

return reversed

def main():

input\_string = "Hello, world!"

reversed\_string = reverse\_string(input\_string)

print(f"Reversed string: {reversed\_string}")

if \_\_name\_\_=="\_\_main\_\_":

main()

Corrected Overall Code:

def reverse\_string(s):

reversed = ""

for i in range(len(s)-1,-1,-1):

reversed += s[i]

return reversed

def main():

input\_string = "Hello, world!"

reversed\_string = reverse\_string(input\_string)

print(f"Reversed string: {reversed\_string}")

if \_\_name\_\_=="\_\_main\_\_":

main()

Explanation :

No Error

**Objective: To identify and fix errors in a Python program that validates user input.**

Code 2:

def get\_age():

age = input("Please enter your age: ")

if age.isnumeric() and age >= 18:

return int(age)

else:

return None

def main():

age = get\_age()

if age:

print(f"You are {age} years old and eligible.")

else:

print("Invalid input. You must be at least 18 years old.")

if \_\_name\_\_=="\_\_main\_\_":

main()

Corrected Overall Code:

def get\_age():

age = input("Please enter your age: ")

if age.isnumeric() and int(age) >= 18:

return int(age)

else:

return None

def main():

age = get\_age()

if age:

print(f"You are {age} years old and eligible.")

else:

print("Invalid input. You must be at least 18 years old.")

if \_\_name\_\_=="\_\_main\_\_":

main()

Error :

**if age.isnumeric() and age >= 18:**

Output : Please enter your age: 18

ERROR!

Traceback (most recent call last):

File "<string>", line 16, in <module>

File "<string>", line 9, in main

File "<string>", line 3, in get\_age

TypeError: '>=' not supported between instances of 'str' and 'int'

Corrected Code :

**if age.isnumeric() and int(age) >= 18:**

Output: Please enter your age: 18

You are 18 years old and eligible.

Explanation :

The **‘isnumeric()’** method returns a boolean indicating whether all characters in the string are numeric. However, in the original code, the comparison **‘age >= 18’** is attempting to compare a string to an integer, which would result in a TypeError. To fix this, the corrected code converts the string **‘age’** to an integer before performing the comparison.

**Objective: To identify and fix errors in a Python program that reads and writes to a file.**

Code 3:

def read\_and\_write\_file(filename):

try:

with open(filename, 'r') as file:

content = file.read()

with open(filename, 'w') as file:

file.write(content.upper())

print(f"File '{filename}' processed successfully.")

except Exception as e:

print(f"An error occurred: {str(e)}")

def main():

filename = "sample.txt"

read\_and\_write\_file(filename)

if \_\_name\_\_=="\_\_main\_\_":

main()

Corrected Overall Code:

def read\_and\_write\_file(filename):

try:

with open(filename, 'r') as file:

content = file.read()

except FileNotFoundError:

print(f"Error: File '{filename}' not found.")

return

except Exception as e:

print(f"An error occurred while reading the file: {str(e)}")

return

try:

with open(filename, 'w') as file:

file.write(content.upper())

print(f"File '{filename}' processed successfully.")

except Exception as e:

print(f"An error occurred while writing to the file: {str(e)}")

def main():

filename = "sample.txt"

read\_and\_write\_file(filename)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Error :

**Output: An error occurred: [Errno 2] No such file or directory: 'sample.txt'**

Corrected Code :

**except FileNotFoundError:**

**print(f"Error: File '{filename}' not found.")**

**return**

**except Exception as e:**

**print(f"An error occurred while reading the file: {str(e)}")**

**return**

Output: ERROR!

Error: File 'sample.txt' not found.

Explanation :

This error is not related to any specific coding issue. Its because the **‘sample.txt’** file is not there but if the file is present. It wont show error. The code now is well-improved version of the previous code. It now includes specific error handling for the file reading and writing operations. When the file is not found, it prints an error message and returns from the function. Additionally, it catches and handles any other exceptions that might occur during the file reading and writing processes.

Objective: The code aims to implement the merge sort algorithm. However, there is a bug in the code. When the student runs this code, it will raise an error or produce incorrect output. The student's task is to identify and correct the bug.

Code 4:

def merge\_sort(arr):

if len(arr) <= 1:

return arr

mid = len(arr) // 2

left = arr[:mid]

right = arr[mid:]

merge\_sort(left)

merge\_sort(right)

i = j = k = 0

while i < len(left) and j < len(right):

if left[i] < right[j]:

arr[k] = left[i]

i += 1

else:

arr[k] = right[j]

j += 1

k += 1

while i < len(left):

arr[k] = left[i]

i += 1

k += 1

while j < len(right):

arr[k] = right[j]

j += 1

k += 1

arr = [38, 27, 43, 3, 9, 82, 10]

merge\_sort(arr)

print(f"The sorted array is: {arr}")

Corrected Overall Code:

def merge\_sort(arr):

if len(arr) <= 1:

return arr

mid = len(arr) // 2

left = arr[:mid]

right = arr[mid:]

# Recursively sort the left and right subarrays

left = merge\_sort(left)

right = merge\_sort(right)

i = j = k = 0

while i < len(left) and j < len(right):

if left[i] < right[j]:

arr[k] = left[i]

i += 1

else:

arr[k] = right[j]

j += 1

k += 1

while i < len(left):

arr[k] = left[i]

i += 1

k += 1

while j < len(right):

arr[k] = right[j]

j += 1

k += 1

return arr

arr = [38, 27, 43, 3, 9, 82, 10]

merge\_sort(arr)

print(f"The sorted array is: {arr}")

Error :

**There was no error.**

Corrected Code :

**# Recursively sort the left and right subarrays**

**left = merge\_sort(left)**

**right = merge\_sort(right)**

. Explanation :

The recursive calls to **‘merge\_sort’** for the **‘left’** and **‘right’** subarrays are now assigned back to ‘left’ and **‘right’**.The sorted subarrays are returned from each recursive call, ensuring that the sorting results are captured.The corrected code now correctly sorts the array using the merge sort algorithm.