Assignment – 1

 # 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()
```

Output:

Reversed string: !dlroW ,olleH

Explanation:

- There are no errors in the above code. It has been executed successfully.
- It defines a function reverse_string that takes a string as input and returns reversed string.
- The main function then uses this reverse_string function to reverse the string "Hello, World!" and prints the result.

 # Code 2:

Explanation: The error is because the age variable is returned as a string from the get_age() function, but you are comparing it with an integer in the if

statement. (>=) comparison operator between a string and an integer is not allowed in Python, which could lead to a TypeError.

The corrected code is:

```
def get_age():
    age = int(input("Please enter your age: "))
    if age >= 18:
        return 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()
```

Output-1

```
Please enter your age:  23
You are 23 years old and eligible.
```

Output-2

```
Please enter your age:  15
Invalid input. You must be at least 18 years old.
```

 # Code – 3:

The corrected code:

```
def read_and_write_file(filename):
```

```
    try:
        # Step 1: Read the content of the file
        with open(filename, 'r') as file:
            content = file.read()
        # Step 2: open and write the modified content
        with open(filename, 'w') as file:
        # Step 3: write the content in uppercase to the file
            file.write(content.upper())
        # Step 4: print a success message
        print(f"File '{filename}' processed successfully.")
    except Exception as e:
        # Step 5: If any error occurs, print an error message
        print(f"An error occurred: {str(e)}")
def main():
    # Step 6: Set the filename to be processed
    filename = "sample.txt"
    # Step 7: Call the read_and_write_file function
    read_and_write_file(filename)
# Step 8: Call the main function if the script is executed as the main program
if __name__ == "__main__":
    main()
```

Explanation:

- The output of the given code would depend on the content of the "sample.txt" file and whether the code executes successfully.
- It reads the content of the "sample.txt" file.
- It then writes the uppercase version of the content back to the same file.

Output:

```
File 'sample.txt' processed successfully.
```

If, for example, the "sample.txt" file contains the text "Hello, World!", after running the code, the content of the file would be changed to "HELLO, WORLD!".

 # Code – 4:

The corrected 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 halves
    left = merge_sort(left)
    right = merge_sort(right)

    i = j = k = 0

    # Merge the sorted halves back into the original array
    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  # Fix: return the sorted array

arr = [38, 27, 43, 3, 9, 82, 10]
merge_sort(arr)
print(f"The sorted array is: {arr}")
```

Explanation:

- The issue is that the merge_sort function is not returning any value.
- By explicitly returning arr at the end of the merge_sort function, we ensure that the sorted array is correctly propagated through the recursive calls.

Output:

The sorted array is: [3, 9, 10, 27, 38, 43, 82]