Here’s a line-by-line explanation of your code, formatted as a detailed breakdown that you can include in your GitHub repository:

try:

x = int(input("Enter the number to check for palindromic: "))

1. **try:**
   * This starts a try block where the code will attempt to execute the statements inside it. If an error occurs, it will jump to the corresponding except block.
2. **input("Enter the number to check for palindromic: ")**
   * Prompts the user to input a number with the message: "Enter the number to check for palindromic".
3. **int(input(...))**
   * Converts the user input (a string) into an integer. If the input is not a valid integer (e.g., the user enters letters), Python raises a ValueError.

str1 = str(x)

1. **str(x)**
   * Converts the integer x into a string. This is necessary because we need to manipulate the number as a sequence of characters to check if it's a palindrome.

str2 = str1[::-1]

1. **str1[::-1]**
   * This slices the string str1 in reverse order. The slicing syntax [::-1] means:
     + Start at the end of the string.
     + Move backward by one character at a time.
     + Continue until the beginning of the string is reached.
   * The result is the reversed version of str1, stored in str2.

if str1 == str2:

print("The given number is palindromic.")

1. **if str1 == str2:**
   * Compares the original string (str1) with its reversed version (str2). If they are equal, it means the number is a palindrome.
2. **print("The given number is palindromic.")**
   * Displays a message indicating that the number is a palindrome.

else:

print("The number is not palindromic.")

1. **else:**
   * Executes this block if the condition str1 == str2 is False (i.e., the number is not a palindrome).
2. **print("The number is not palindromic.")**
   * Displays a message indicating that the number is not a palindrome.

except ValueError:

print("Invalid input! Please enter a valid integer.")

1. **except ValueError:**
   * This block handles any ValueError exceptions raised in the try block. A ValueError occurs if the user inputs something that cannot be converted into an integer (e.g., letters, symbols).
2. **print("Invalid input! Please enter a valid integer.")**
   * Displays an error message to the user, prompting them to enter a valid integer.

**Summary:**

This program:

1. Prompts the user to input a number.
2. Checks if the number is a palindrome by comparing its original and reversed forms.
3. Handles invalid inputs (non-integer values) gracefully by displaying an error message.

**Example Run:**

1. **Input:** 121  
   **Output:** "The given number is palindromic."
2. **Input:** 123  
   **Output:** "The number is not palindromic."
3. **Input:** abc  
   **Output:** "Invalid input! Please enter a valid integer."