**Working of the functions in the Python Script**

The python script provides a command-line interface for the user to interact with the Windows Registry.

It performs read, write, deletion of individual entries, and deletion of entire keys under the HKEY\_CURRENT\_USER ***for the Zoom software registry path.***

1. *Reading a Registry Entry*
   1. Parameters: key\_path & value\_name
   2. It opens the specified registry key and queries the value. If successful, it prints the value.
2. *Writing a Registry Entry*
   1. Parameters: key\_path, value\_name & value\_data.
   2. It opens the specified registry key and sets the value. The data is converted to a string format and written to the registry.
3. *Deleting a Registry Entry*
   1. Parameters: key\_path
   2. The dentry function deletes a specified value from the registry.
4. *Deleting a Registry Key*
   1. Parameters: key\_path
   2. The dkey function deletes an entire registry key.
5. *Main Function:* The main function provides a user interface to choose between the above operations. It loops until the user decides to exit, ensuring that multiple operations can be performed sequentially.

**Challenges Encountered and Resolutions**

* *Permission Issues:* Working with the registry requires administrative privileges. The script handles WindowsError exceptions and prompts the user to run the script with appropriate permissions if necessary.
* *Data Type Handling:* Handling different data types (strings vs. integers) posed a challenge. The script converts all data to strings before writing to the registry to simplify the handling process.
* *Error Handling:* The script uses multiple exception handlers (FileNotFoundError, WindowsError, and general Exception) to provide specific feedback for various error scenarios.

**How the Script Handles Different Error Scenarios**

* *File Not Found:* When a specified registry key or value does not exist, the script catches FileNotFoundError and informs the user that the entry was not found.
* *Permission Denied:* If the script encounters permission issues, it catches WindowsError and advises the user to run the script with administrative privileges.
* *General Errors*: For any other unexpected errors, the script catches Exception and prints a generic error message along with the error details. This ensures that the script does not crash unexpectedly.
* *Invalid User Input:* The script prompts the user to enter valid options and handles invalid input gracefully by informing the user and re-prompting for input.

import winreg as wrg

def read(key\_path, value\_name):

try:

location = wrg.HKEY\_CURRENT\_USER

soft = wrg.OpenKeyEx(location, key\_path)

value = wrg.QueryValueEx(soft, value\_name)

wrg.CloseKey(soft)

print(f"Value of '{value\_name}' in '{key\_path}': {value[0]}")

except FileNotFoundError:

print(f"Entry '{value\_name}' not found in '{key\_path}'.")

except WindowsError as e:

print(f"Error reading entry: {e.strerror}")

except Exception as e:

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

def write(key\_path, value\_name, value\_data):

try:

location = wrg.HKEY\_CURRENT\_USER

soft = wrg.OpenKeyEx(location, key\_path, 0, wrg.KEY\_SET\_VALUE)

wrg.SetValueEx(soft, value\_name, 0, wrg.REG\_SZ, str(value\_data))

wrg.CloseKey(soft)

print(f"Entry '{value\_name}' added to '{key\_path}' successfully.")

except WindowsError as e:

print(f"Error writing entry: {e.strerror}")

except Exception as e:

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

def dentry(key\_path, value\_name):

try:

location = wrg.HKEY\_CURRENT\_USER

soft = wrg.OpenKeyEx(location, key\_path, 0, wrg.KEY\_SET\_VALUE)

wrg.DeleteValue(soft, value\_name)

wrg.CloseKey(soft)

print(f"Entry '{value\_name}' deleted from '{key\_path}' successfully.")

except FileNotFoundError:

print(f"Entry '{value\_name}' not found in '{key\_path}'.")

except WindowsError as e:

print(f"Error deleting entry: {e.strerror}")

except Exception as e:

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

def dkey(key\_path):

try:

location = wrg.HKEY\_CURRENT\_USER

soft = wrg.OpenKeyEx(location, key\_path, 0, wrg.KEY\_SET\_VALUE)

wrg.DeleteKey(soft, "")

wrg.CloseKey(soft)

print(f"Key '{key\_path}' deleted successfully.")

except FileNotFoundError:

print(f"Key '{key\_path}' not found.")

except WindowsError as e:

print(f"Error deleting key: {e.strerror}")

except Exception as e:

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

def main():

while True:

print("Select one of these to perform an action.")

print("1: Read a Registry Entry")

print("2: Write a Registry Entry")

print("3: Delete a Registry Entry")

print("4: Delete a Registry Key")

print("5: Exit")

action = input("Enter the number of the action you want to perform: ")

if action == "1":

key\_path = input("Enter the key path: ")

value\_name = input("Enter the value name: ")

read\_registry\_entry(key\_path, value\_name)

elif action == "2":

key\_path = input("Enter the key path: ")

value\_name = input("Enter the value name: ")

value\_data = input("Enter the value data: ")

write\_registry\_entry(key\_path, value\_name, value\_data)

elif action == "3":

key\_path = input("Enter the key path: ")

value\_name = input("Enter the value name: ")

dentry(key\_path, value\_name)

elif action == "4":

key\_path = input("Enter the key path: ")

dkey(key\_path)

elif action == "5":

print("Exiting the program.")

break

else:

print("Invalid option. Choose b/w 1 - 4.")

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

main()