**OOP PROJECT PRELIMINARY REPORT**

**HANGMAN GAME**

**Introduction:**

Hangman is a popular word guessing game where the player attempts to build a missing word by guessing one letter at a time. After a certain number of incorrect guesses, the game ends and the player lose. The game also ends if the player correctly identifies all the letters of the missing word. The program will be including concepts of Object Oriented Programming.

**Description:**

* The program opens up a console output of the program which contains brief description and instructions of the game.
* The text file contains a dictionary of words that has been previously generated and saved by the user.
* User can add or remove any word by file handling.
* The program then randomly selects the word from the file for the user to guess.
* The program does not executes during runtime the words which are already been used during runtime.
* The program contains features various functions to make the program efficient.

**Goals:**

The goal of this project is to write a graphically based game of hangman using Object Oriented Programming.

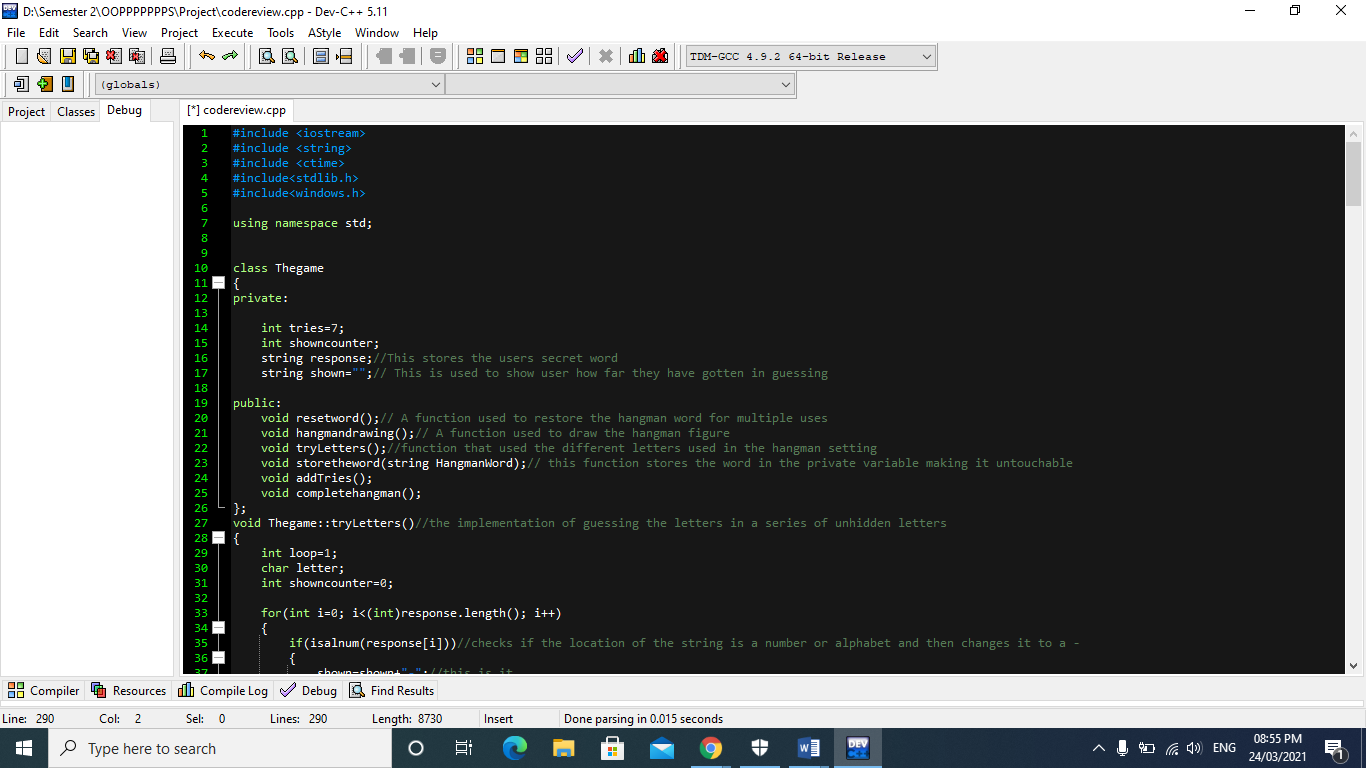
* The program displays the option of single player or 2 players.
* The computer generates a word for the single player while the 2 player game option lets the one of the player decide the word for the other player.
* The number of spaces and dashes are then printed on the screen automatically.
* The player then starts guessing letter. If the letter appears in the word then the computer puts the letter in its proper location. If not then the computer adds one more body part to the hangman. We used 4 body parts (e.g. head, body, arms, and legs).
* If the entire body is completed then the player loses. If the player guesses the word before the body is completed then the player wins.
* After guessing a letter the letter should be removed from the available alphabet so that it is not guessed again.
* The user should be alerted if they try to guess the same letter more than once.

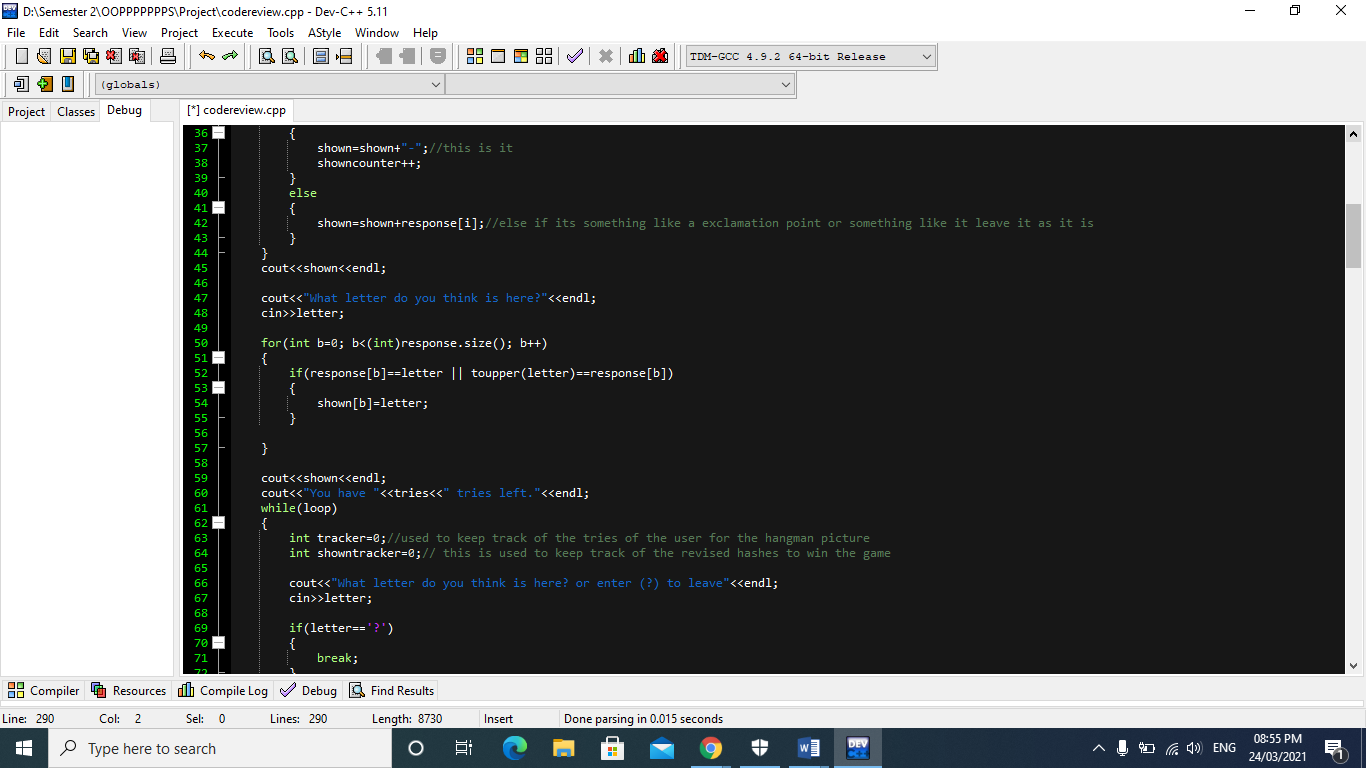
**Problem:**

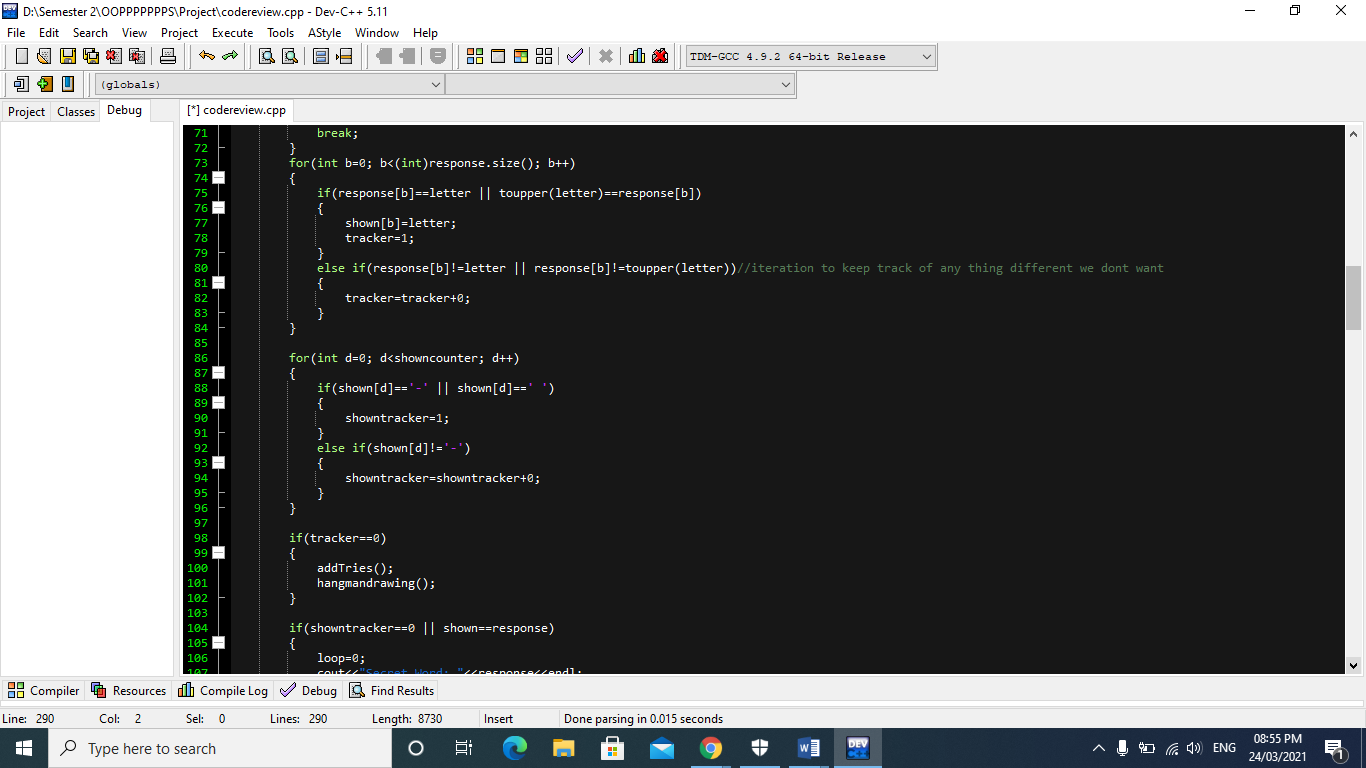
For Hangman, we need to store 3 pieces of information:

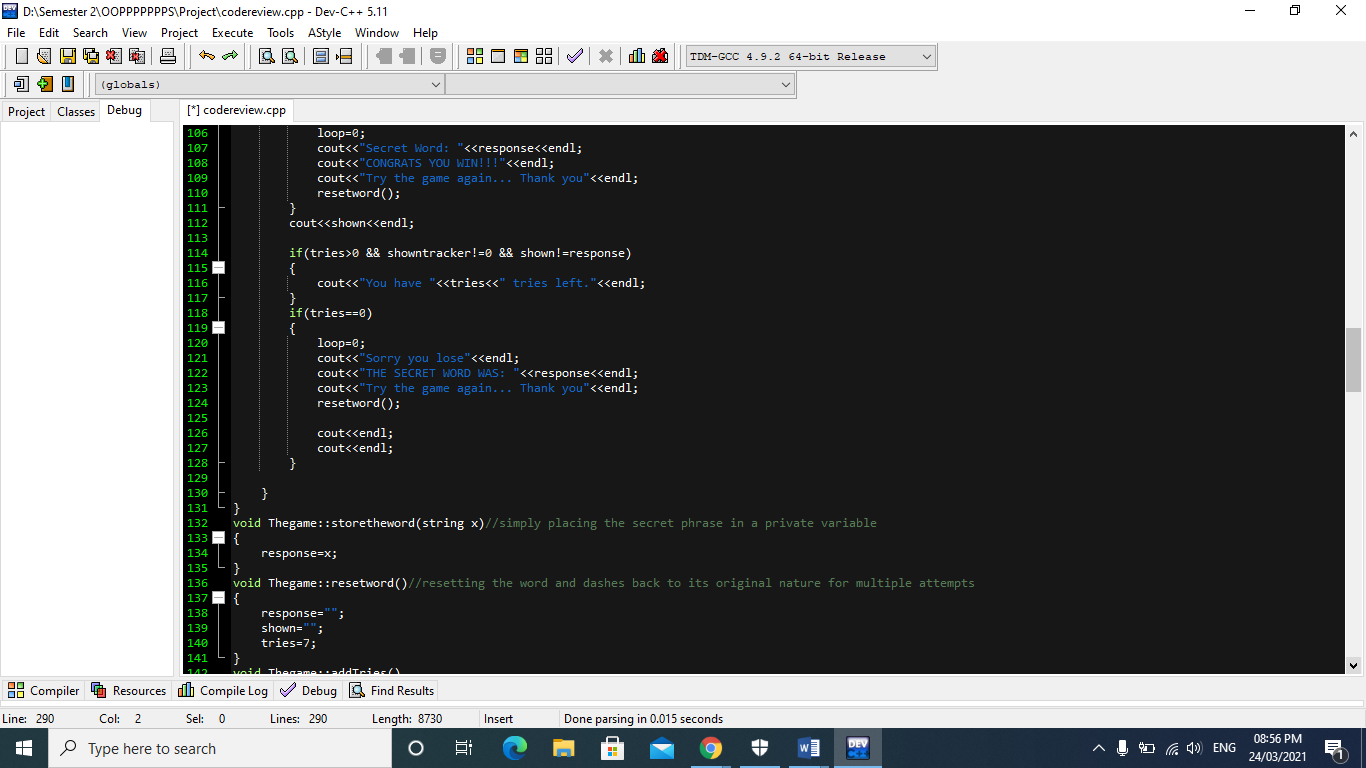
* Secret word: The word to be guessed.
* Letters guessed: The letters been guessed so far.
* Mistakes made: The number of incorrect guesses made so far.

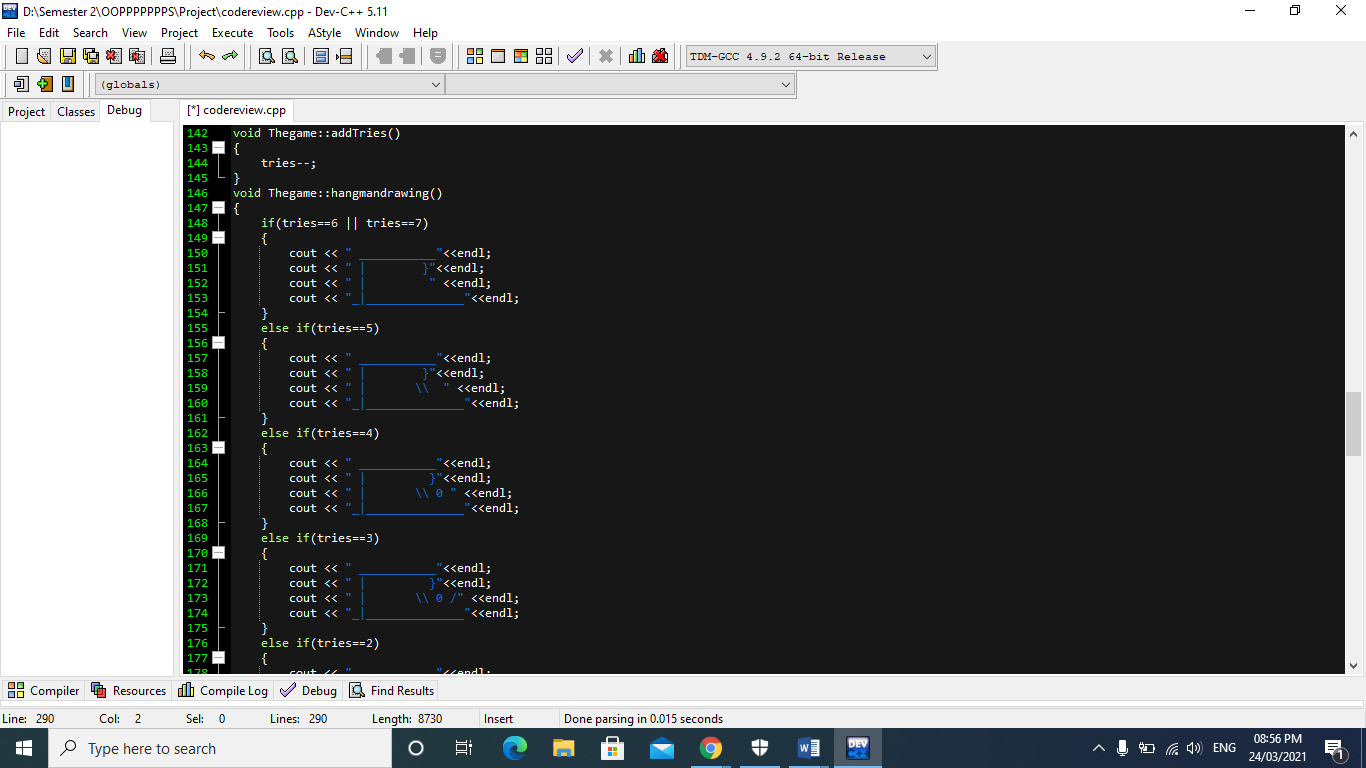
**Existing Solution:**

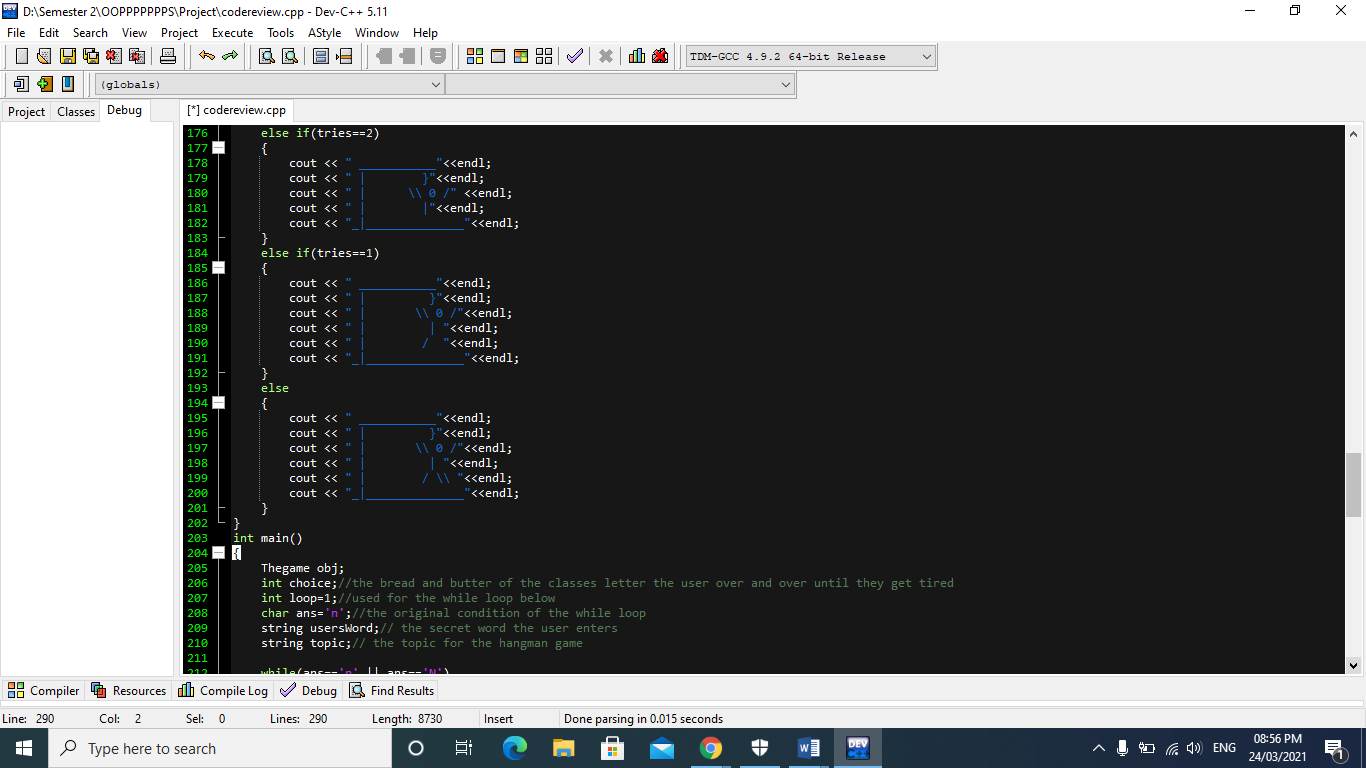


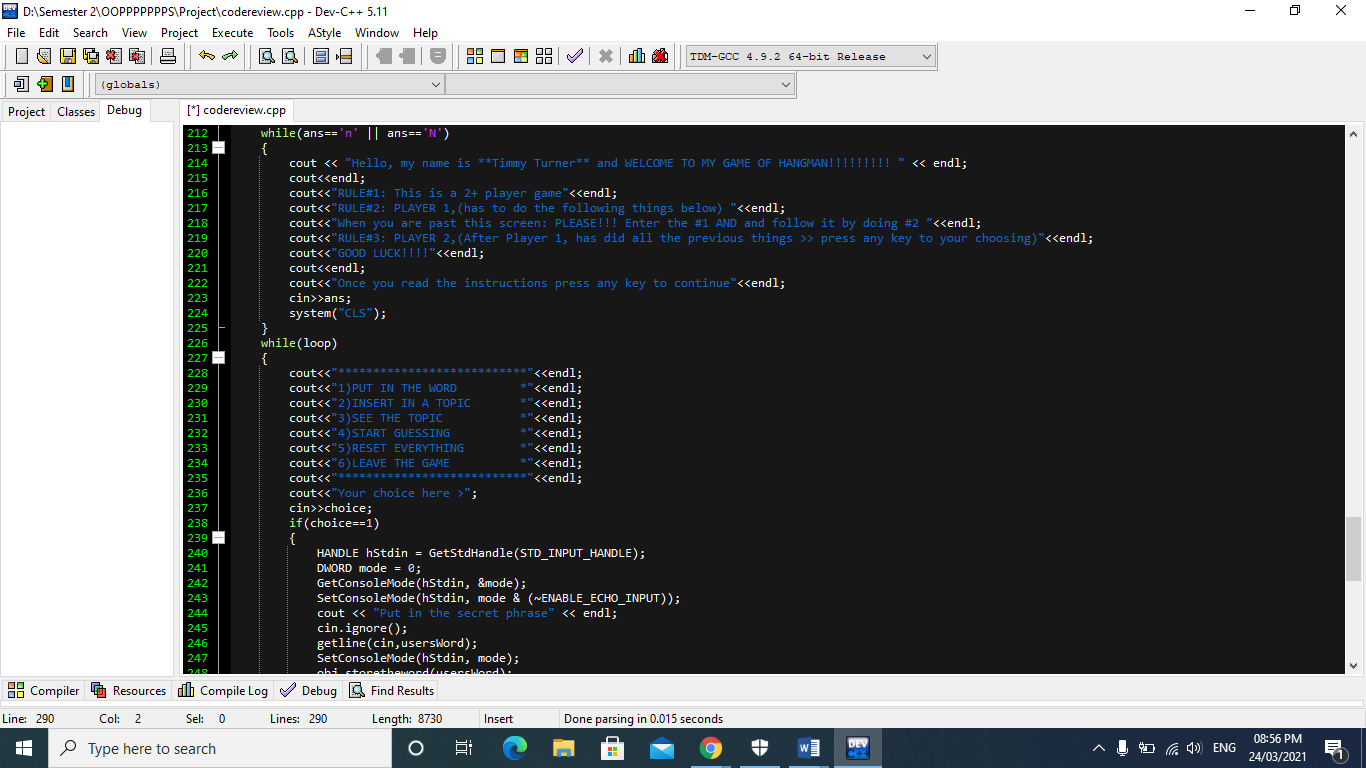


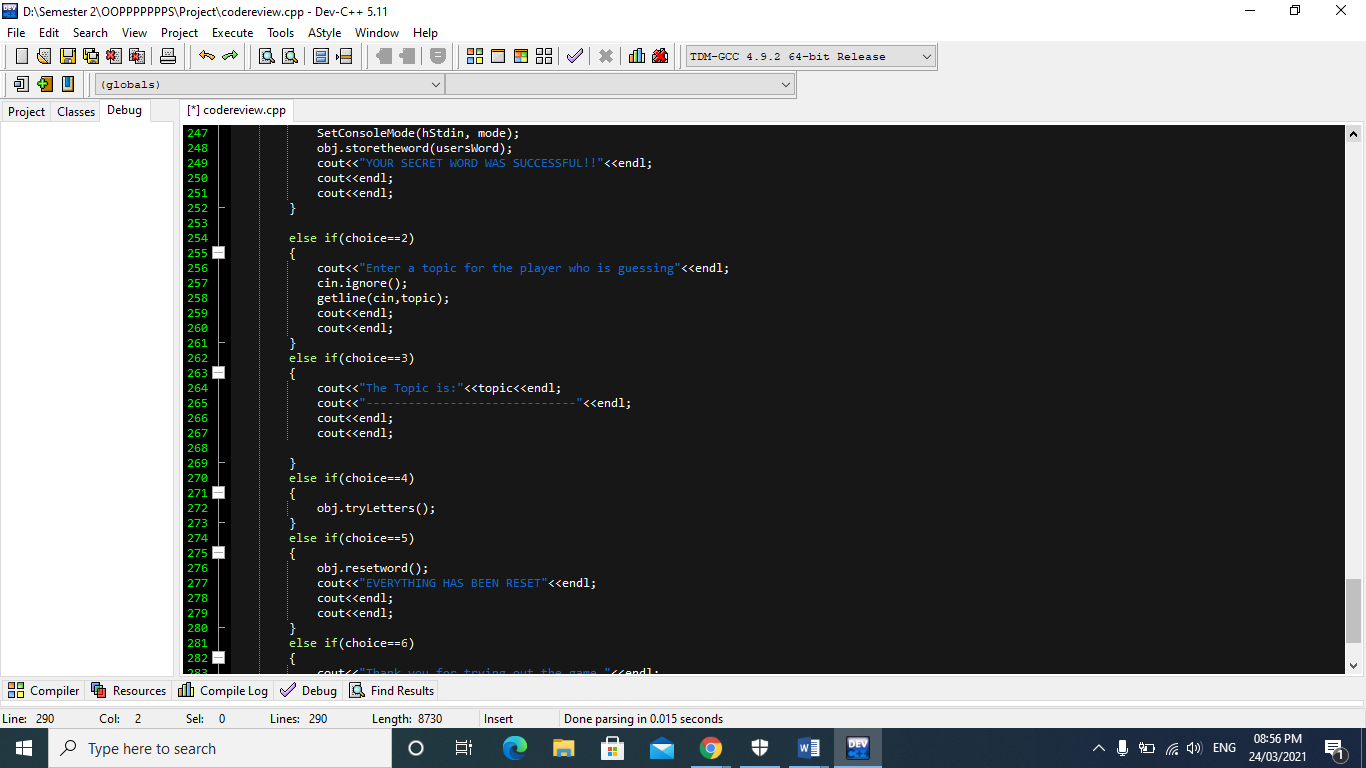


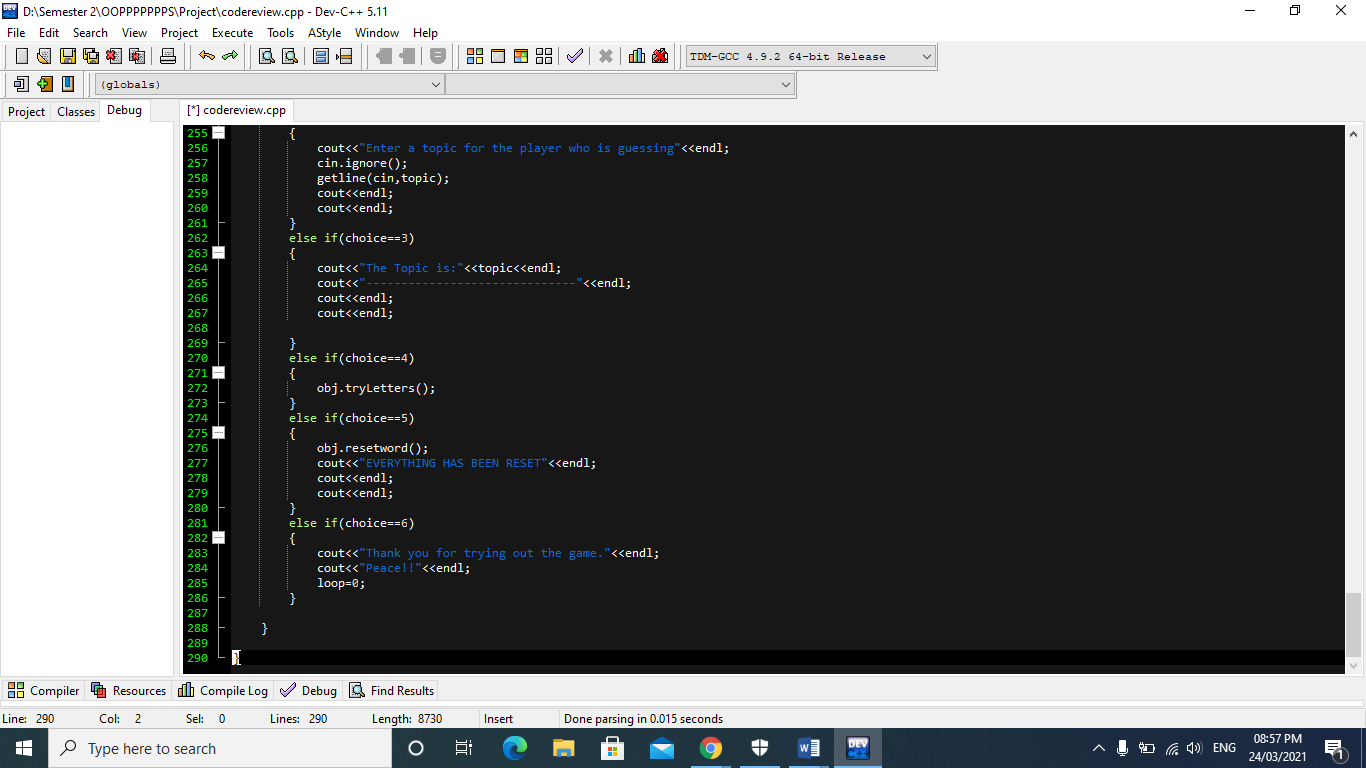












**Class diagram:**

|  |
| --- |
| **Hangman** |
| * tries: int * counter: int * guessedLetters: string * foundLetters: string * missedLetters: string |
| * addGuessed(): void * resetWord(): void * guesses(): void * tries(): void * drawHangman (): void * completeHangman(): void * result(): void |

|  |
| --- |
| **Word** |
| * secretWord: string * wordLength: int * blanks: string |
| * word(): string * setWord(): void * getWord(): string * setBlanks(): void * getBlanks(): string |

**Topics:**

The following concepts will be implemented in this project.

* Classes and objects
* Abstraction and encapsulation
* Constructors and destructors
* Member initialization
* Inline functions
* Data hiding
* Inheritance
* Filing and I/O stream

**Group Members:**

Bismah Akram (20K-0449) – Group Leader

Noor Fatima (20K-0406)

Section: BSCS-D2

**Conclusion:**

The Hangman will be coded using all the concepts of Object Oriented Programming in Visual Studio in C++.