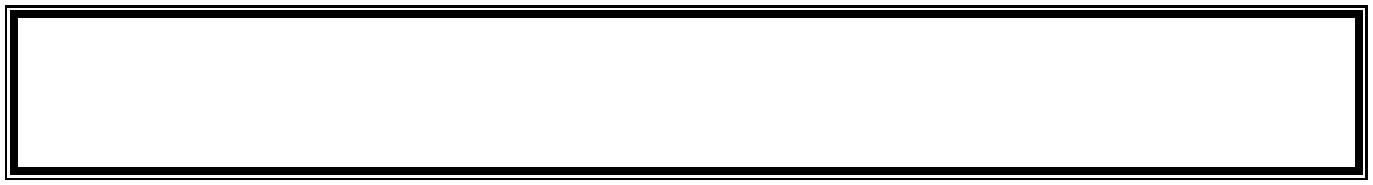
Odd Semester (2016)



**BINUS UNIVERSITY**

**BINUS INTERNATIONAL**



**Assignment Cover Letter**

**(Individual Work)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Information**: | ***Surname*** | ***Given Names*** | ***Student ID Number*** |
| *1.* | Taneka | Archel | 2001585556 |
| *2.* |  |  |  |
| *3.* |  |  |  |
| *4.* |  |  |  |
| *5.* |  |  |  |
| **Course Code** | **:** | **Course Name** | **:** Academic English |
| **Class** | **:** | **Name of Lecturer(s)** | **:** 1. Ida Bagus Kerthyayana Manuaba |
|  |  |  | 2. |
| **Major** | **:** | Computer Science |  |
| **Title of Assignment** | **:** | Code-Type Exercise |  |
| (if any) |  |  |  |
| **Type of Assignment** | **:** | Individual |  |
| **Submission Pattern** |  |  |  |
| **Due Date:** 6th November 2016 |  | **Submission Date** | : 6th November 2016 |

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

**Plagiarism/Cheating**

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

**Declaration of Originality**

By signing this assignment, I understand, accept and consent to Binus International terms and policy on plagiarism. Here with I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

*Signature of Student: (Name of Student)*

*1.*

*2.*

*3.*

*4.*

*5*

CHAPTER I

DESCRIPTION

As stated, the program called Code-Type Exercise. As a Computer Science student, coding is a part of our daily life. Everything that we did in the class, working on assignment or project related to coding. Have you ever encountered an assignment or homework with a brief deadline? How do you overcome it? In this case, we, as a software engineer not only just analyzing a problem quickly but also implement them into the coding section as quick as we did with the analyzing. Quick enough when analyzing is a common thing as a software engineer, but quick in both analyzing and typing the code is the unusual one. So that is why I came up with this idea.

This kind of exercise is different with another typing exercise. You will copy a sample text code exactly the same. This program focus on both typing speed and of course, accuracy. No need to think about algorithm design and logic, just type it as fast as possible with minimum errors. Since the aim of this program is for people who have just started learning the basic programming, there are no such complicated and complex code. Just a simple one, such as how to print “Hello world” to the screen. There are 3 levels, easy (for those who has just started programming), medium (for those who has got the grasp of the basic programming) and hard (for more challenging stuffs).

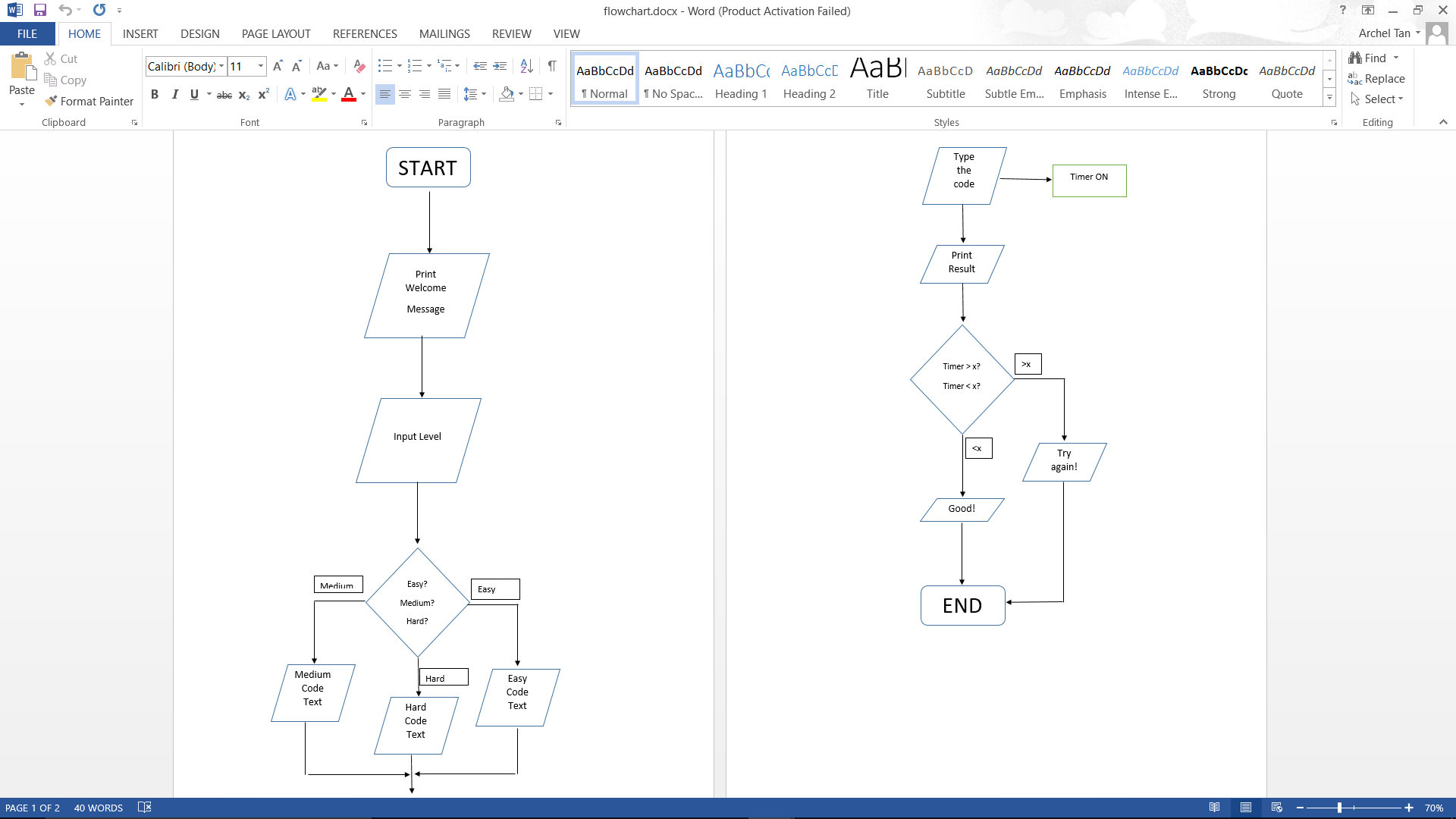
This program will calculate user’s WPM (Words Per Minute), calculate timer on when the user finished typing the sample code and accuracy (the user typed how many incorrect and correct characters). The program then will check if user’s typing speed is faster or slower than the average speed and check for timer whether it’s slower or faster than the average time. If the user is not satisfied with the result, they can redo it with the same level or with different level.

CHAPTER II

DESIGN, PLAN AND EXPLANATIONS

* 1. **Design/Plan**

I decided to use a *flowchart* for helping me designing my idea.

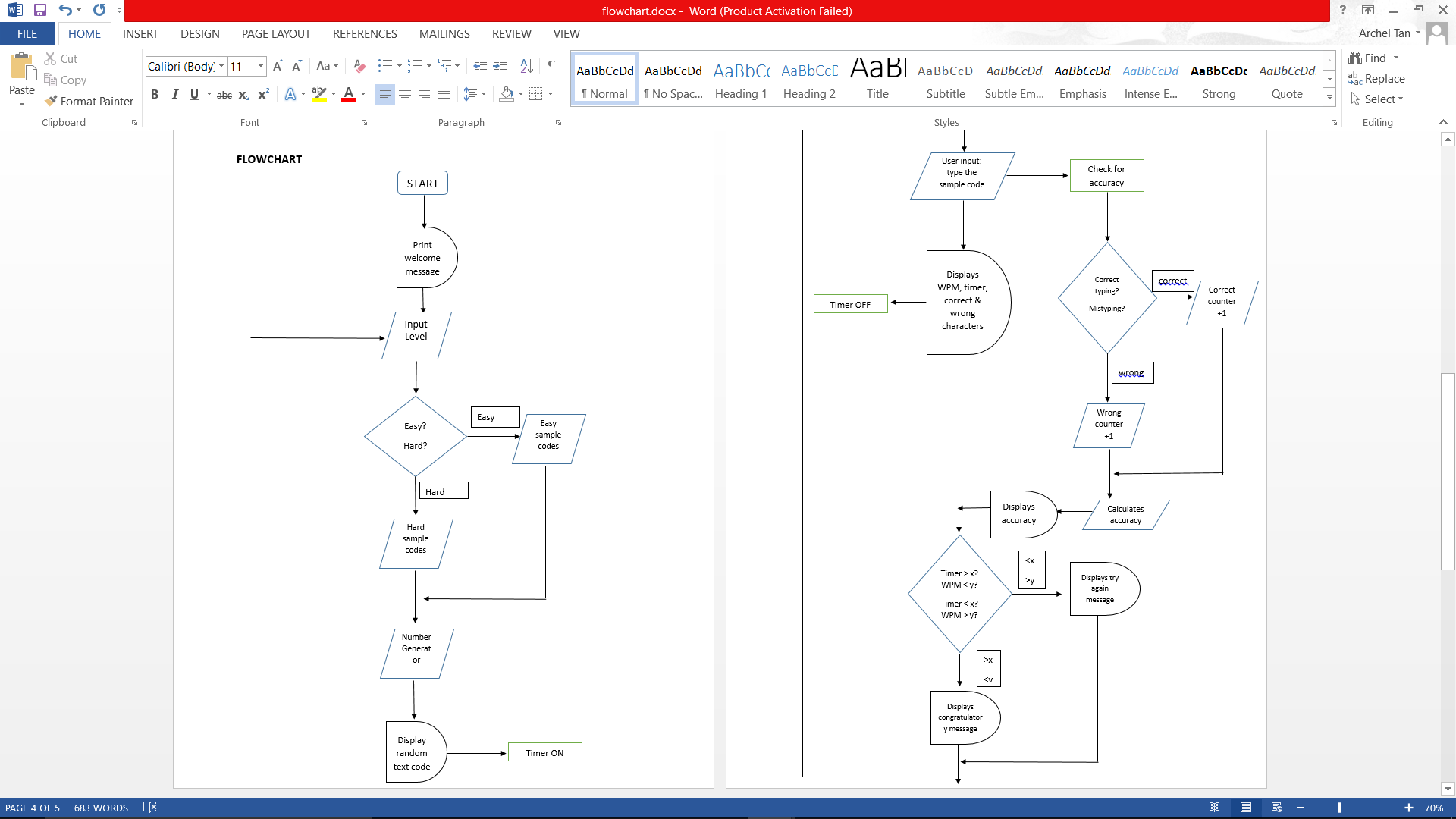
* + 1. **1st Version Flowchart (Date Modified: 7th October 2016)**

This was my first flowchart design. It was a simple flowchart with start and end of program/process, input/output operation and decisions. The program starts and print a welcome message. Then it asks user input for choosing levels provided. According to the level chosen, a sample text code will be displayed to the screen, but something is odd about the level decision part. The display symbol in the flowchart above still incorrect. A display symbol is using a rectangle with round and pointed tip at the both sides. Second thing, decisions are usually followed by 2 output possibilities, but here there were 3 possible outputs. It’s something unusual to have 3 possible outputs from a decision. After level input, a sample text code will be shown to the display screen. Each level only has one sample text code, so there are only 3 sample texts code at the moment. At the same time, the timer also starts.

The timer will stop if the user has finished typing the sample code. It then checks for timer and typing speed. If the timer is exceeding the maximum time provided, a ‘try again’ message will be displayed, same rule applies for checking typing speed, but slightly different logic. If user’s typing speed is exceeding the maximum typing speed provided, a congratulatory message will be displayed. It means that the user typed faster than the average person did. Checking for typing speed and timer is on one declaration for efficiency and more practical than checking it one by one.

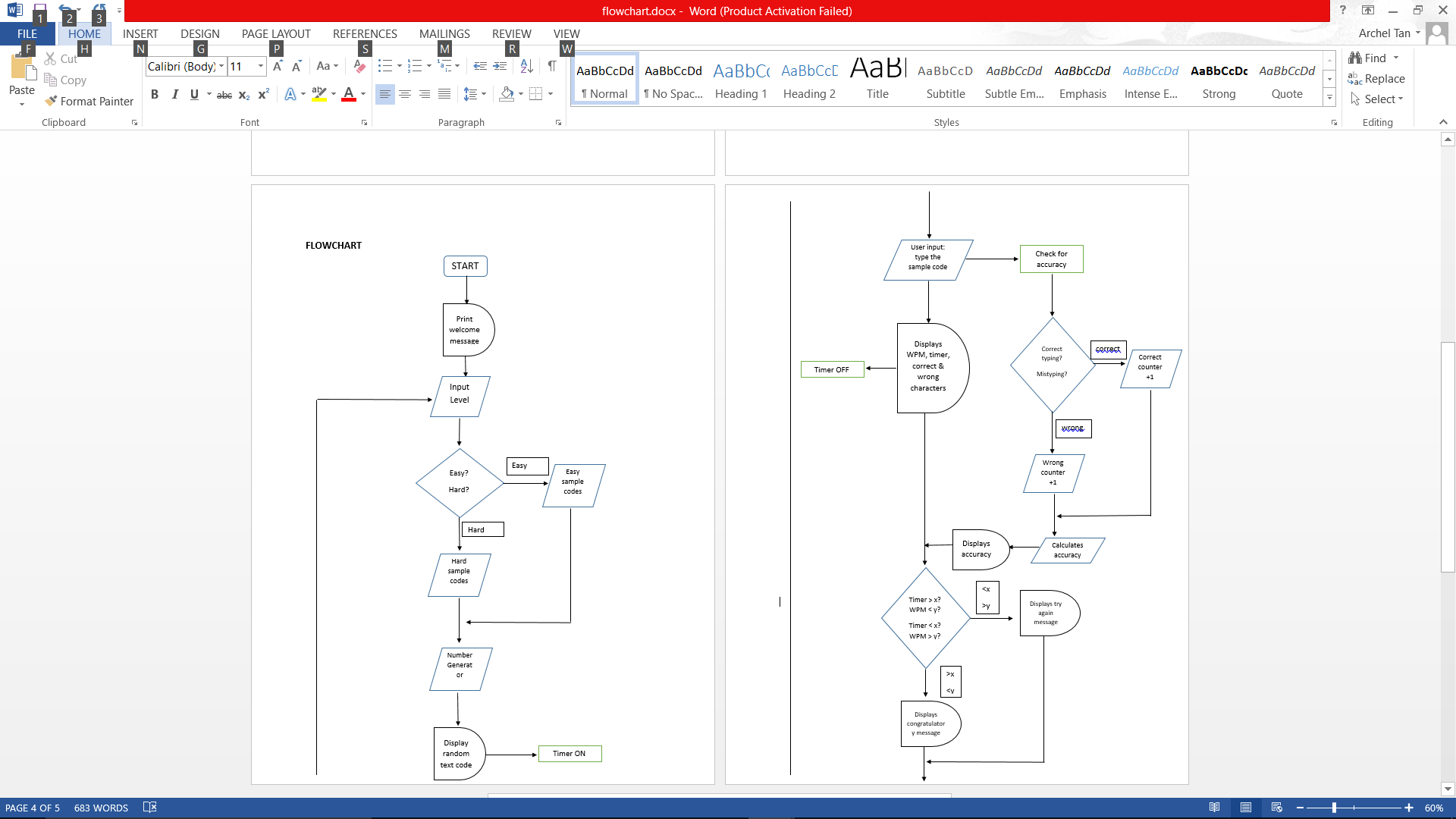
* + 1. **2nd Version Flowchart (Date Modified: 4th November 2016)**

Part 1

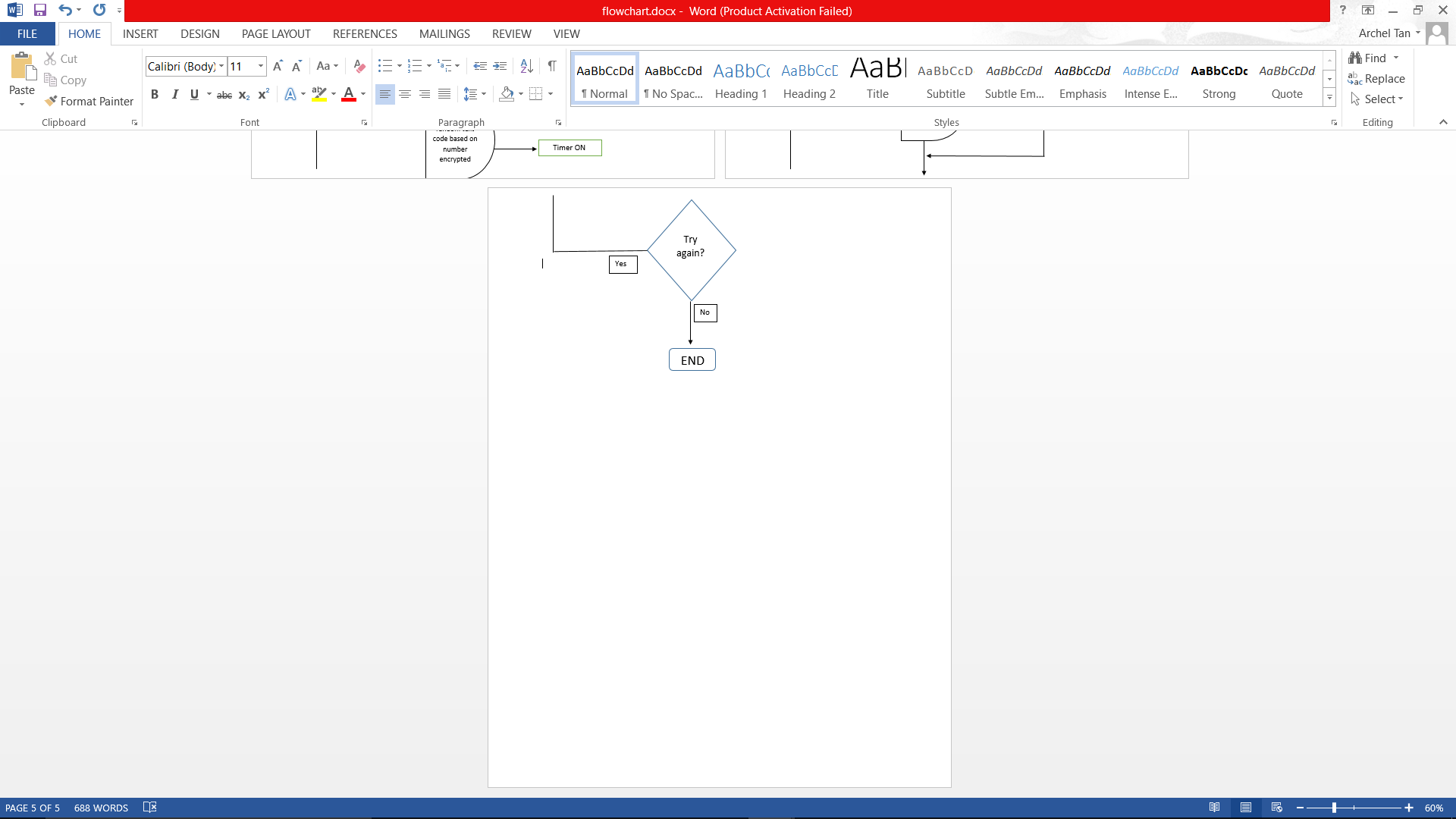


The first part of the flowchart is almost the same one as the 1st version, but there were changes and addition made. Now, we can see that the ‘display’ part has changed with the appropriate symbol. The ‘choosing level’ decision part now makes sense with only 2 possible outputs. There was also an addition about a random number generator. The 1st version flowchart only provided 1 sample text code for each level, but here there were 3 sample texts code in each level, and it displayed randomly based on the number encrypted, and at the same time, the timer is turned on.

Part 2



The flowchart continues to the second part, which is the user input (code typing) and display the results. After the program displayed random sample text code based on the level input, user can start typing the sample code given. Here, the process was divided into two. When user is typing, the program also checks for the number of correct typings and mistypings. If any correct typing occurred, the number of correct typing is incremented by 1, same rule applied with the number of mistypings. After user has done the typing, the program will calculate for accuracy and display it simultaneously with typing speed, timer and the number of correct and wrong typings. Then it enters the results checking. It displays whether the user is fast enough to type the code or not. Again, the display symbol was changed from the 1st version.

 Part 3

This is the last part of the flowchart. There was one decision symbol to ask user if they want to do the exercise again, either with the same level, or different level. If the user wants to do it again, then the program will execute the input level process through the end of user input. If not, then the program terminates here.

* 1. **Explanation of each function / class in the design**

There are 7 functions and 1 class used in this program:

* + 1. **Functions**
       1. **Main Function**

This is the main part of the program. Other function calling is also declared here, it also displays a welcome message to the user. Most of the program are running in the main function.

* + - 1. **getUserLevel Function**

This function will ask the user to input the level. There are 3 levels provided, easy, medium and hard. In this function also the description of each level will be displayed to the screen. The return type of this function is void, because it only displays description and does not return anything.

* + - 1. **displayCode Function**

Mainly, this function is to display random sample texts code. There is ‘random number support’ with keyword *srand.* This function is just like a ‘database’ for the program. It sets all of the data including the sample texts code to display it to the screen. This function also calls the getTypeUser and getResult function, this function type is also void, since that there are no thing to return.

* + - 1. **getTypeUser Function**

This is where the user input for typing the code is. There is also ‘timer support’ for calculating the timer. This function will also read input files to check for numbers of correct and mistypings. At the end of the program, this function also display the results such as, typing speed, timer, number of correct typings and mistypings and also accuracy showed in percentage. The return type of this function is double, since this function takes the word parameter which means that every sample text code has different total numbers of words (characters). So, every calculation conducted by this function depends on the words from sample texts code.

* + - 1. **getResult Function**

This is the function for checking the results with the minimum standard. If user’s typing speed and timer is below the standard, then a ‘try again’ message will be displayed to encourage user to practice more. If user’s typing speed and timer surpasses the minimum standard, then a congratulatory message will be displayed and encourage user to try the next level. The return type of this function is void, since it has nothing to return with.

* + - 1. **getRandNum Function**

The ‘random number generator’ is declared in this function. As we can see that there are 2 parameters accept integer data type and it does return a value. The getRandNum function will set 2 numbers which refer to the minimum and maximum number we wish to randomize.

* + - 1. **tryAgain Function**

This function simply asks the user if they want to do the exercise again or not. It will run at the end of the program. The program will go back to the getUserLevel function if the user wishes to do it again or the program will terminate if the user does not want to do it again.

* + 1. **Classes**
       1. **Account Class**

This class simply stores nicknames of the user. At the beginning of the program, it will ask for user’s nickname input. It then will be displayed at the end of the program along with the stats (typing speed, accuracy, timer, numbers of correct and mistypings).

CHAPTER III

TRIALS & ERRORS

* 1. **Trials**

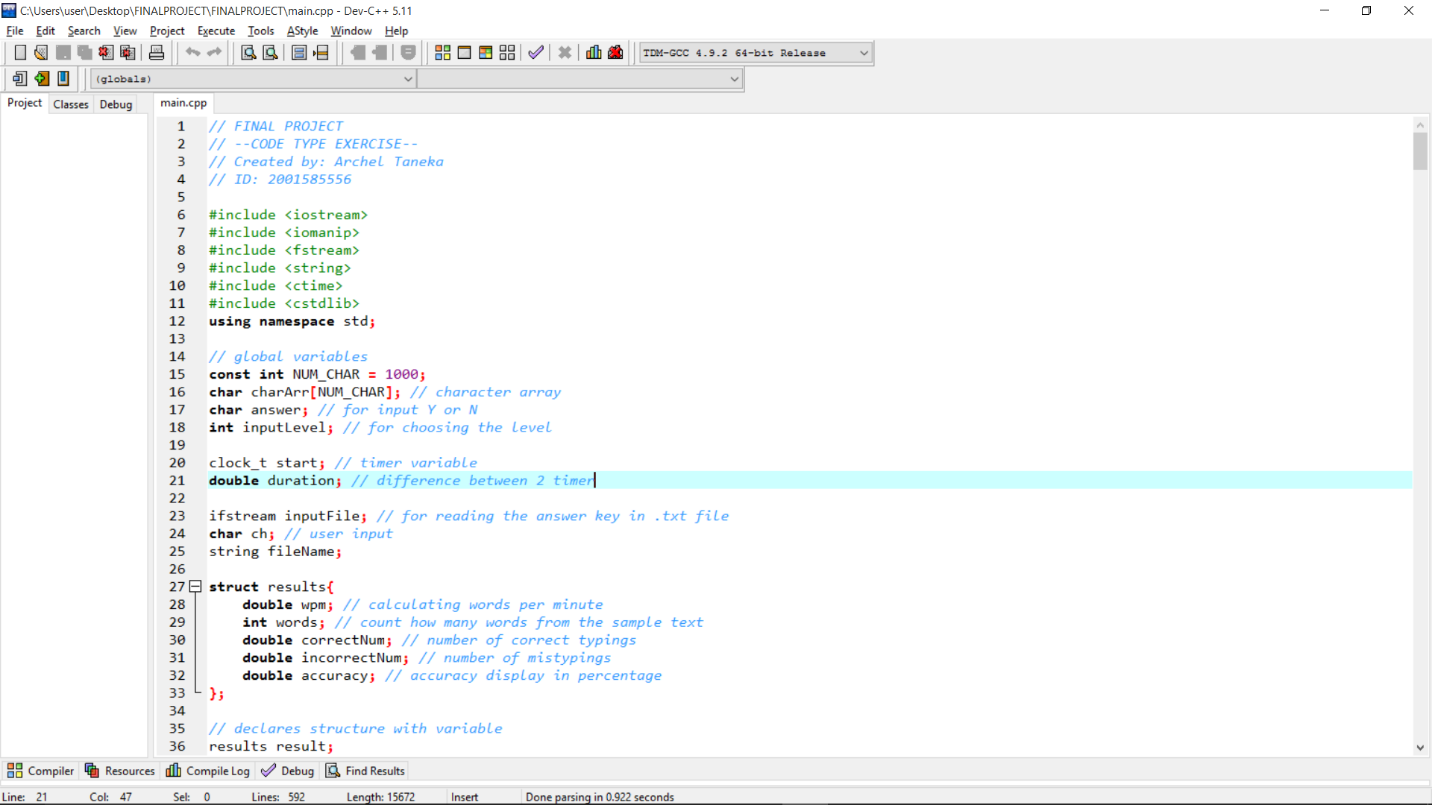
While working on this project, I have also learnt many things not just about coding but also soft skills, such as time management, hardwork, persistent, and also striving for the best results. Without we realize it, working on the final project is like a ‘simulation’ for us in the world of a software engineer, because it is not just about working on your own. Honestly, I could not finish my project without the help my facilitators and colleagues. So, let’s get down to the details.

* + 1. **Coding Section**

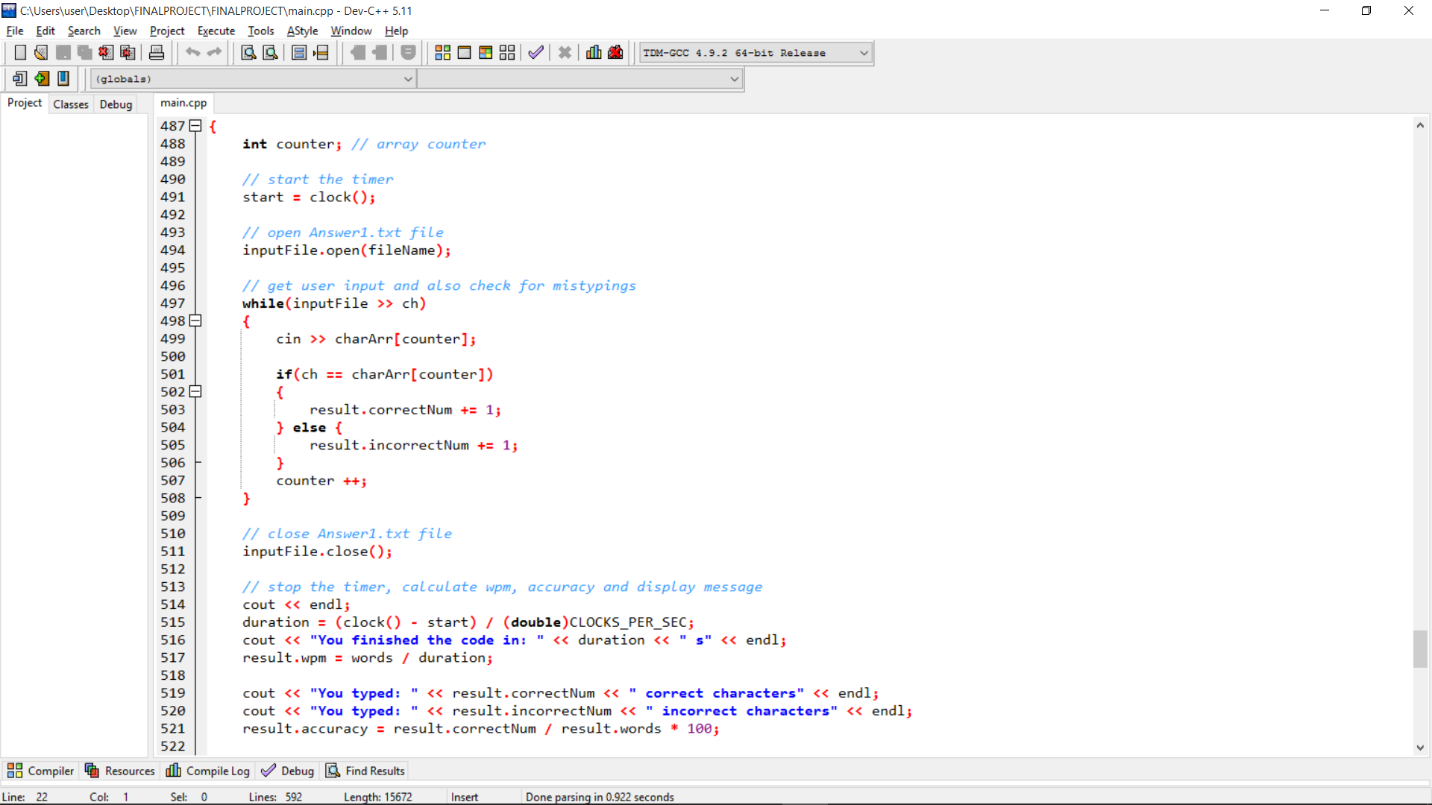
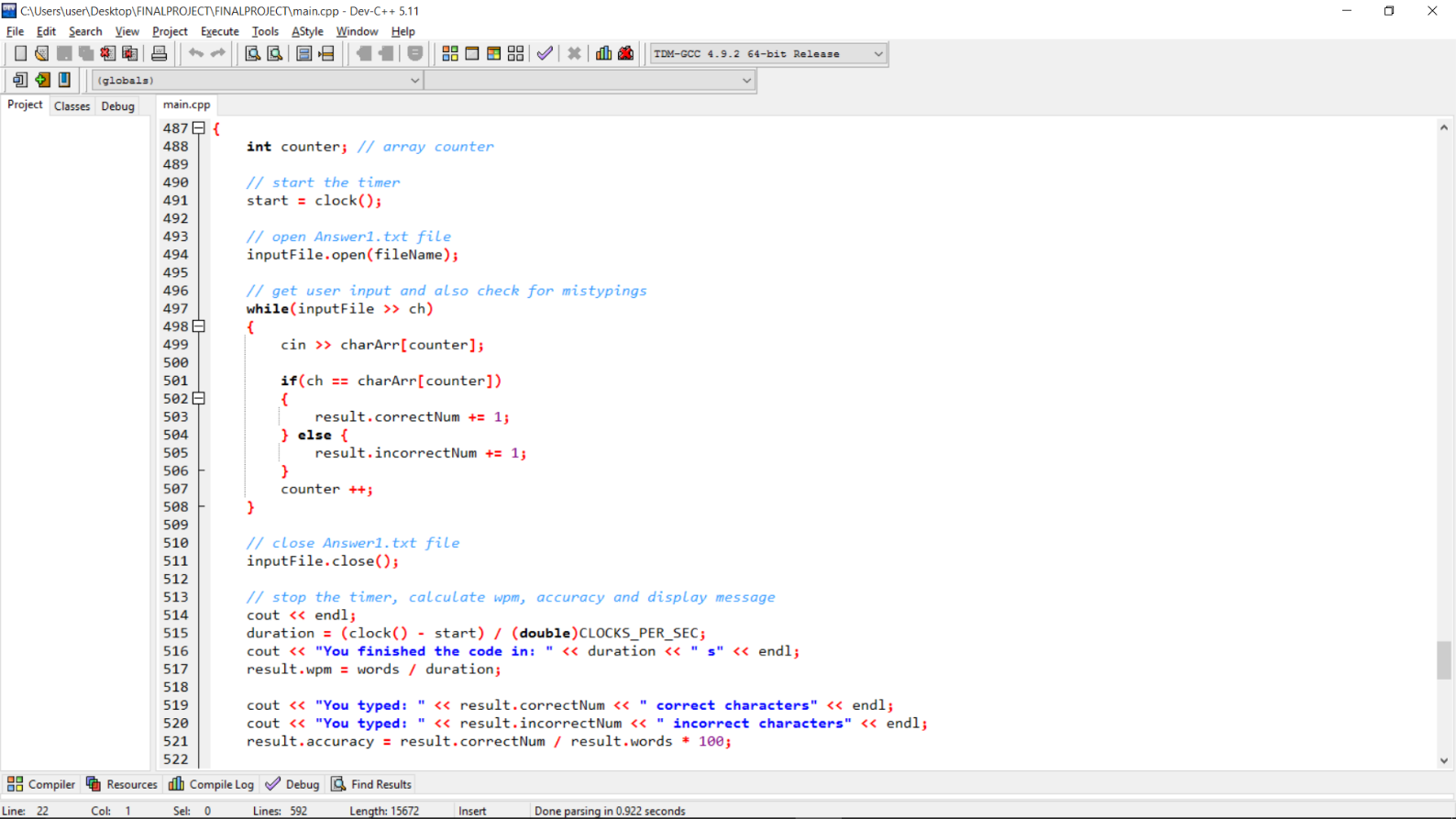
There are a couple of things that I have learnt, where mostly I didn’t get these in a formal class, instead I looked up to the internet and had a consultation with my facilitators.

* + - 1. **Include Time**

How to include time is one of the most crucial part in my final project. As I have mentioned before that after the user has finished the typing, a timer will be displayed to the screen. How many seconds elapsed when they have finished typing.

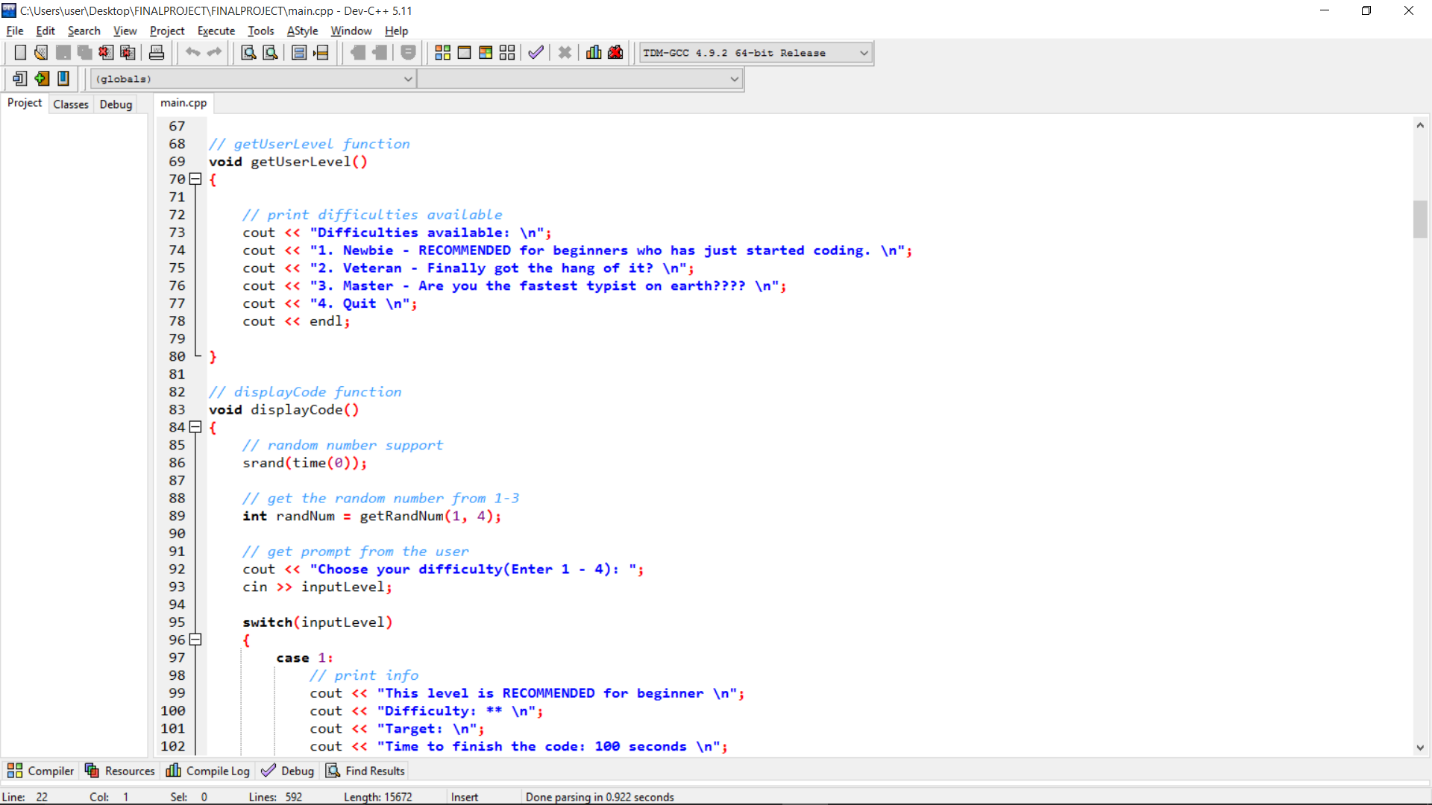
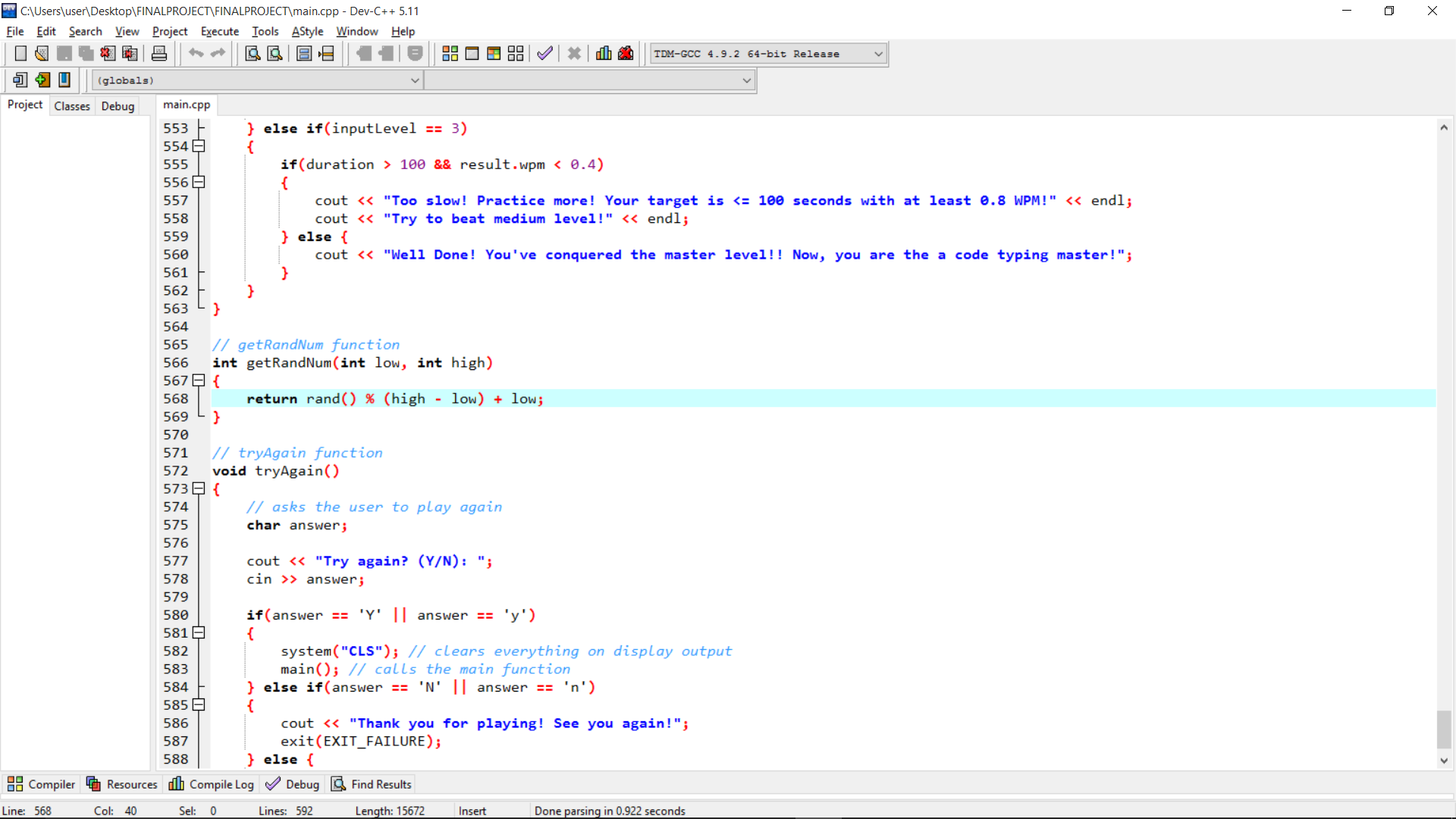


As we can see that there is a keyword ‘clock\_t’. This keyword was declared in global variables, outside the main function and other functions.



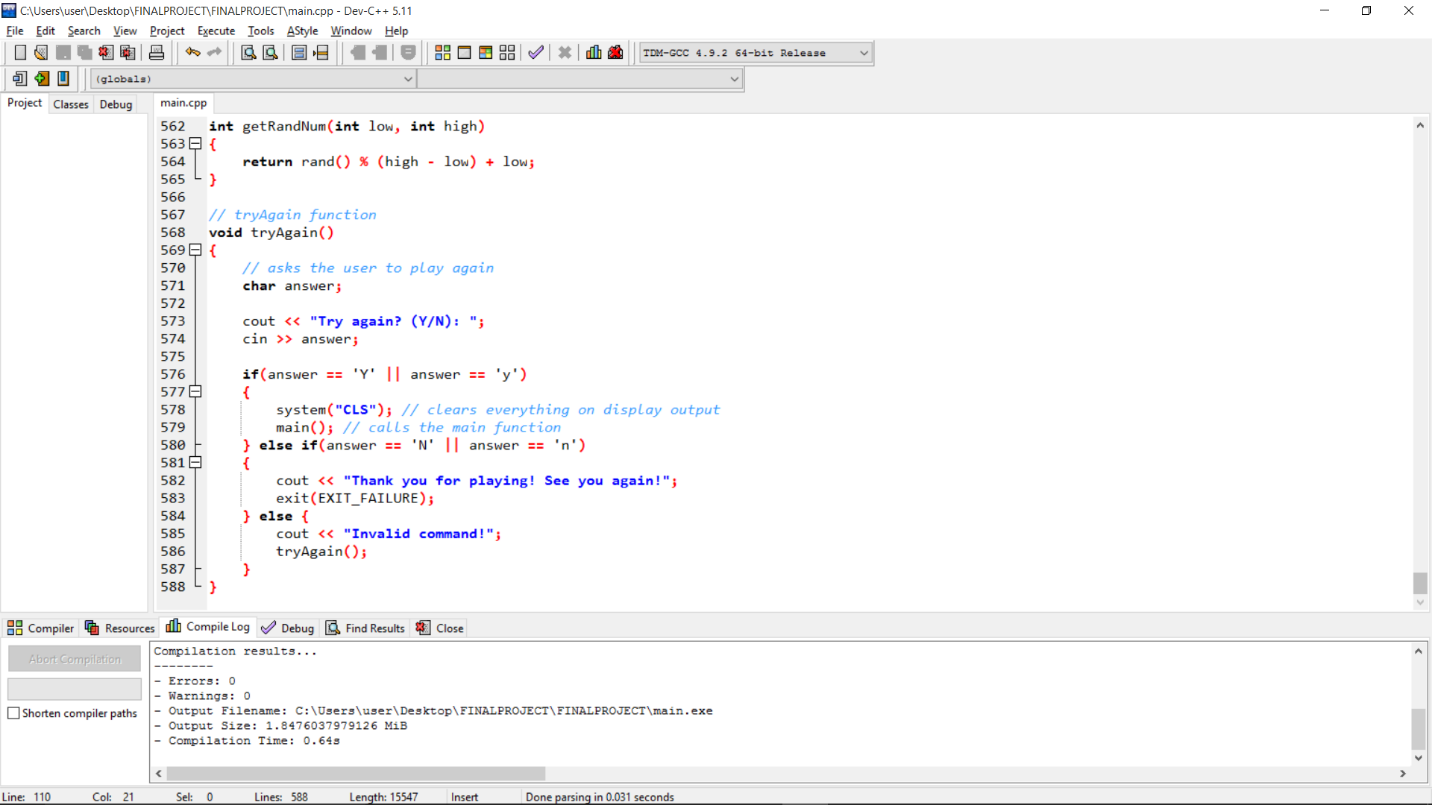
In fact, those pieces of code is for including a time and calculate time elapsed as a timer works. There will be a more explanation about these in Chapter IV.

* + - 1. **Get Random Numbers**

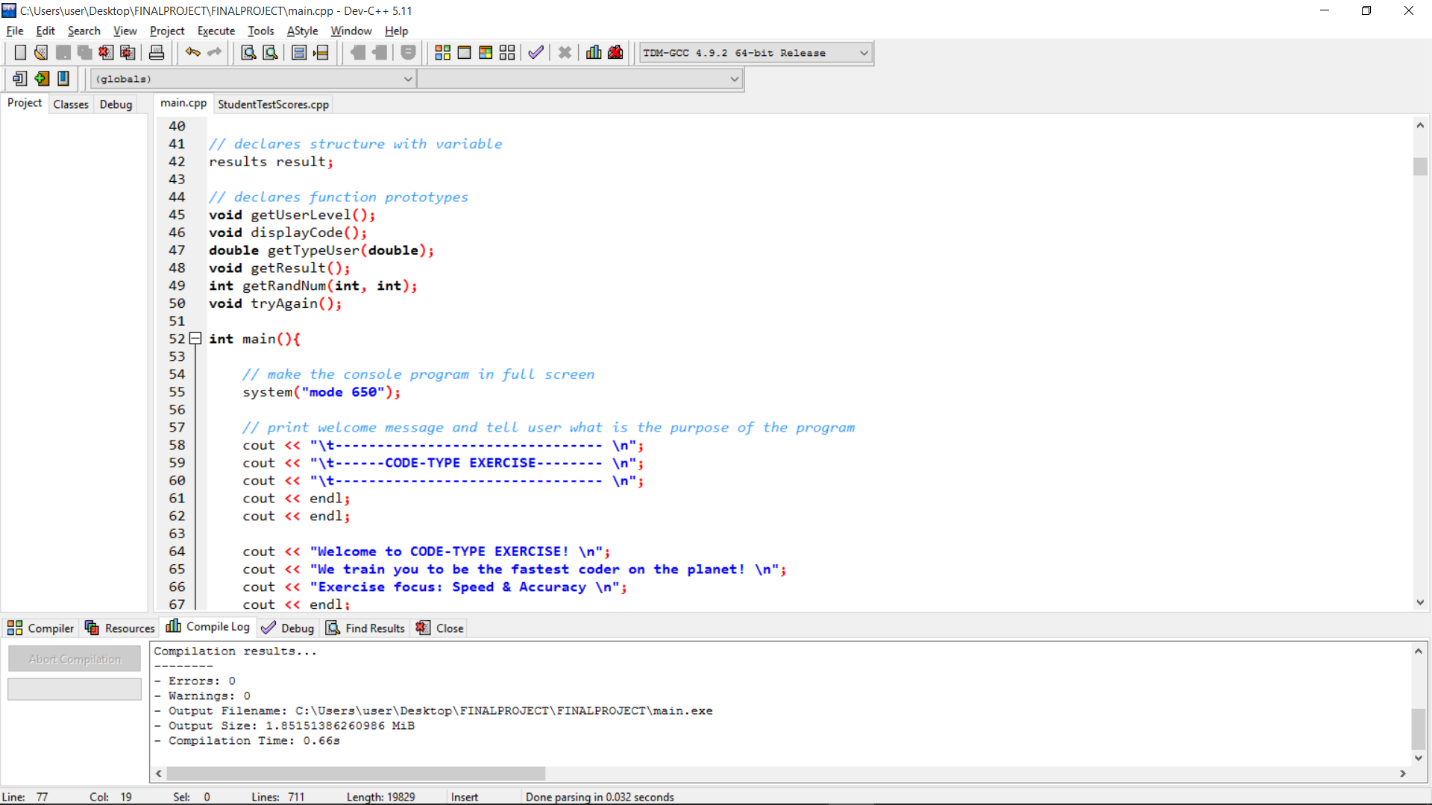


If you still remember, the sample texts code are displayed in an encrypted or in a randomized way, which means that I have to find a solution how to produce a number in a randomized way. The keyword ‘srand’ is the crucial part for randomizing a number. There will also a more explanation about this.

* + - 1. **Using cstdlib Library**

Cstdlib library is generally a header that defines several general purpose functions, including dynamic memory management, random number generation, communication with environment, integer arithmetics, searching, sorting and converting. In this case, I only use cstdlib library to use the command called ‘system’.

There is a command system(“CLS”), this command can be run by including the library cstdlib. Moreover, we can make the console program in a full screen with system(“mode 650”)



* + 1. **Soft Skills**

I believe that soft skills are also take a part of what I get through completing this final project unintentionally. We have to change our mindset, we are no longer an unstable teenager, but as a fine worker in the future, as a software engineer indeed. In my opinion, these are what it takes to be a reliable software engineer.

* + - 1. **Time Management**

Time management is the most important thing when dealing with a project/assignments. How we manage our time, dividing between our daily life. When do we work on the assignments and when to enjoy our free time with families and friends. A good time manager, always keep in mind about doing things little by little, so when it comes near the due date, we simply avoid overwork and get enough rest for the next day. The results are also different. Working on an assignment near the due date makes our creativity and workspace limited, because no time to think about. So, be wise using the time.

* + - 1. **Hardwork**

This is a must for every software engineer to have. A hardworking person and always strive for the best makes someone surrounded by positive vibes. Think of every way and different perspective will surely generate the best results. Ask for the others help if we’re stuck with the problem. Find any possible solutions you’ve ever thought about it before and hopefully God will also helps us achieving our goal.

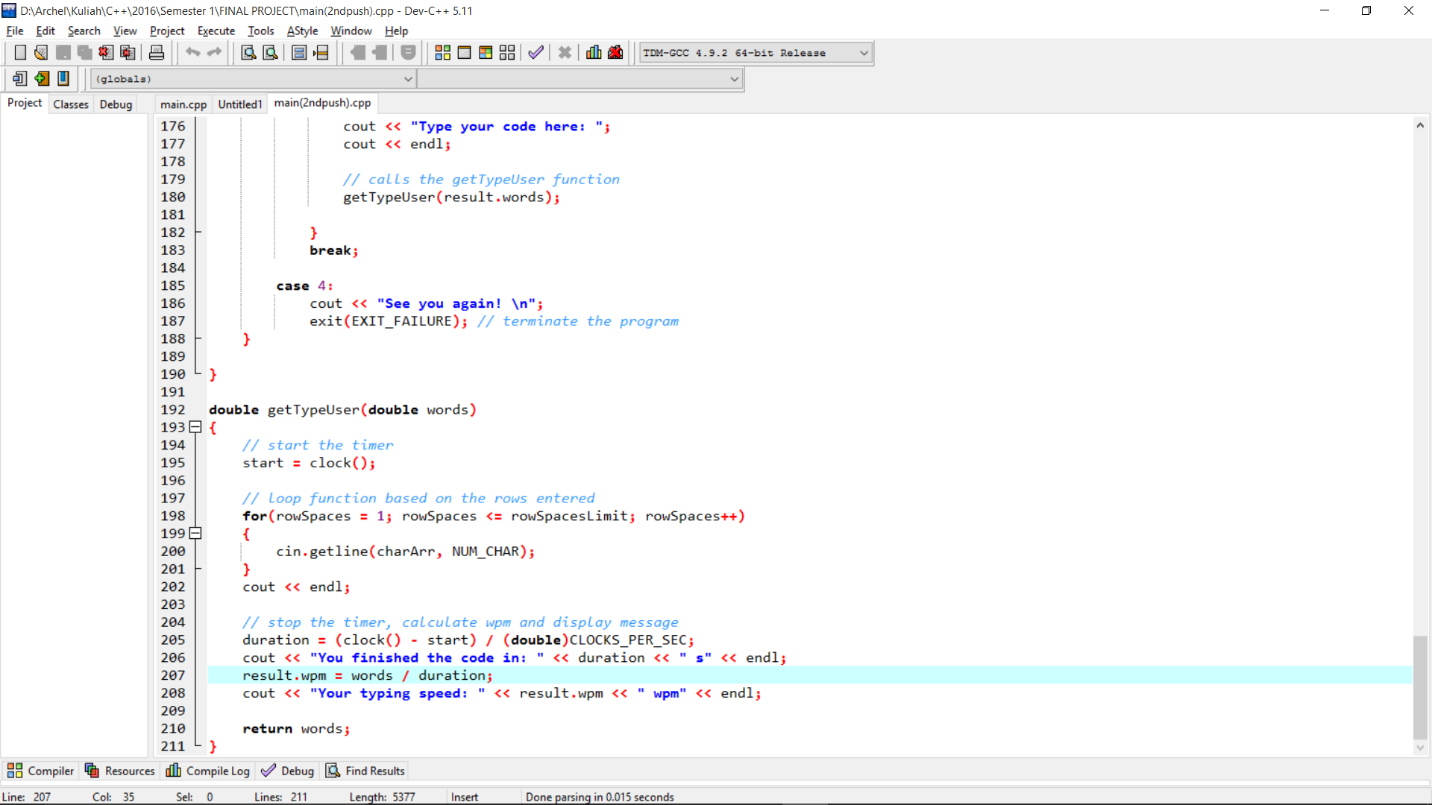
* + - 1. **Helping each other**

When we’re done with our assignments, don’t forget to give a hand with colleagues. Sharing something that we know is a good thing. “Learning by doing” is suitable for this. We don’t need to read books to learn, but by teaching someone, we’ve already learnt something.

* 1. **Errors and Problems Overcame**

In this project, many kind problems I have encountered. Once again I could overcame them is not because on my own knowledge and skill, but with the help of my lecturers and colleagues. In this section, I will share the problems in both syntax and logical errors.

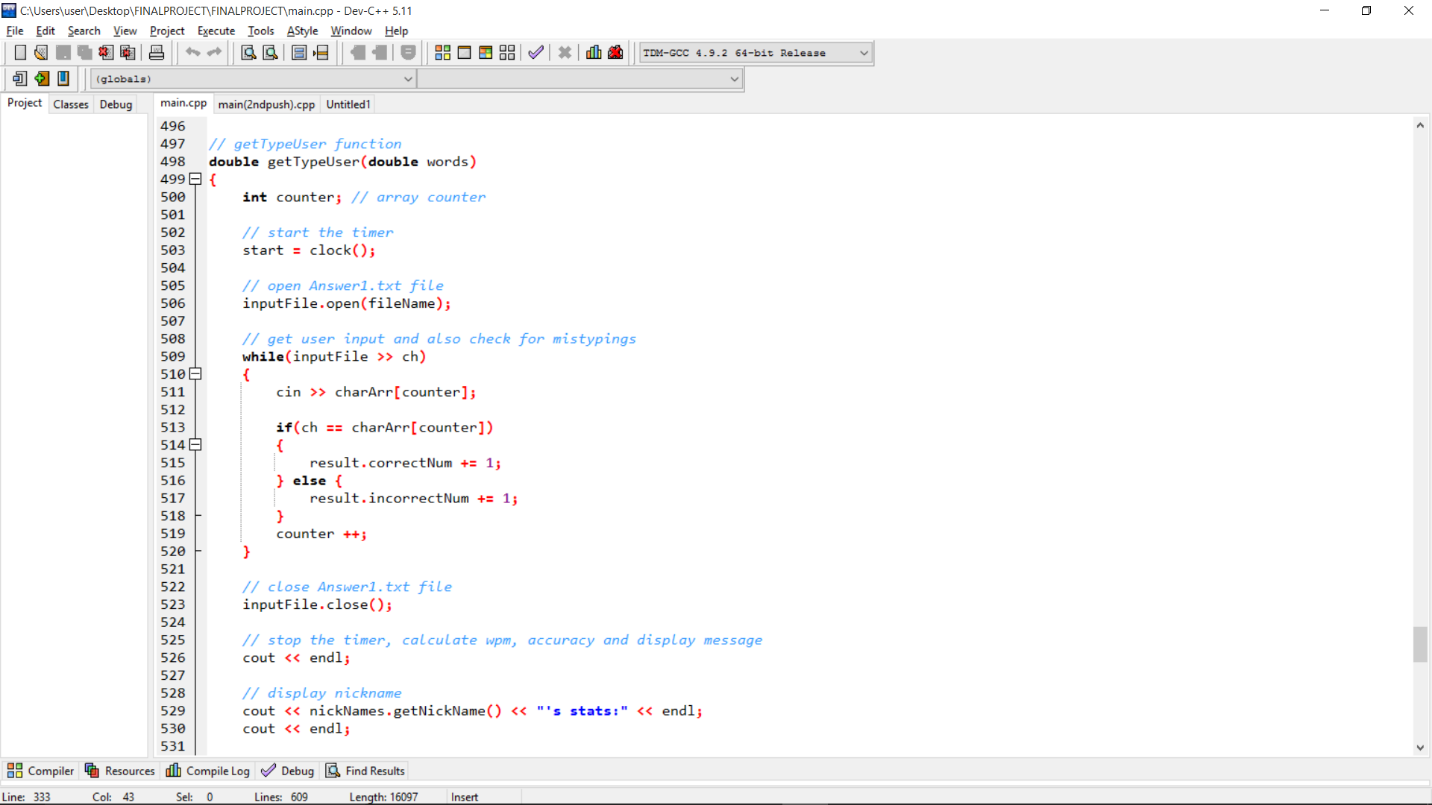
* + 1. **User Input and Checking for Correct & Mistyping**



This was my first method about user input (typing the sample text code section). As we could see that I used the for loop function and array of characters with cin.getline method. The for loop was used to check how many row spaces, in other words how many times user has input the enter key. For example, we want to type this kind of code

#include <iostream>  
using namespace std;   
int main(){  
cout << “hello world”;  
}

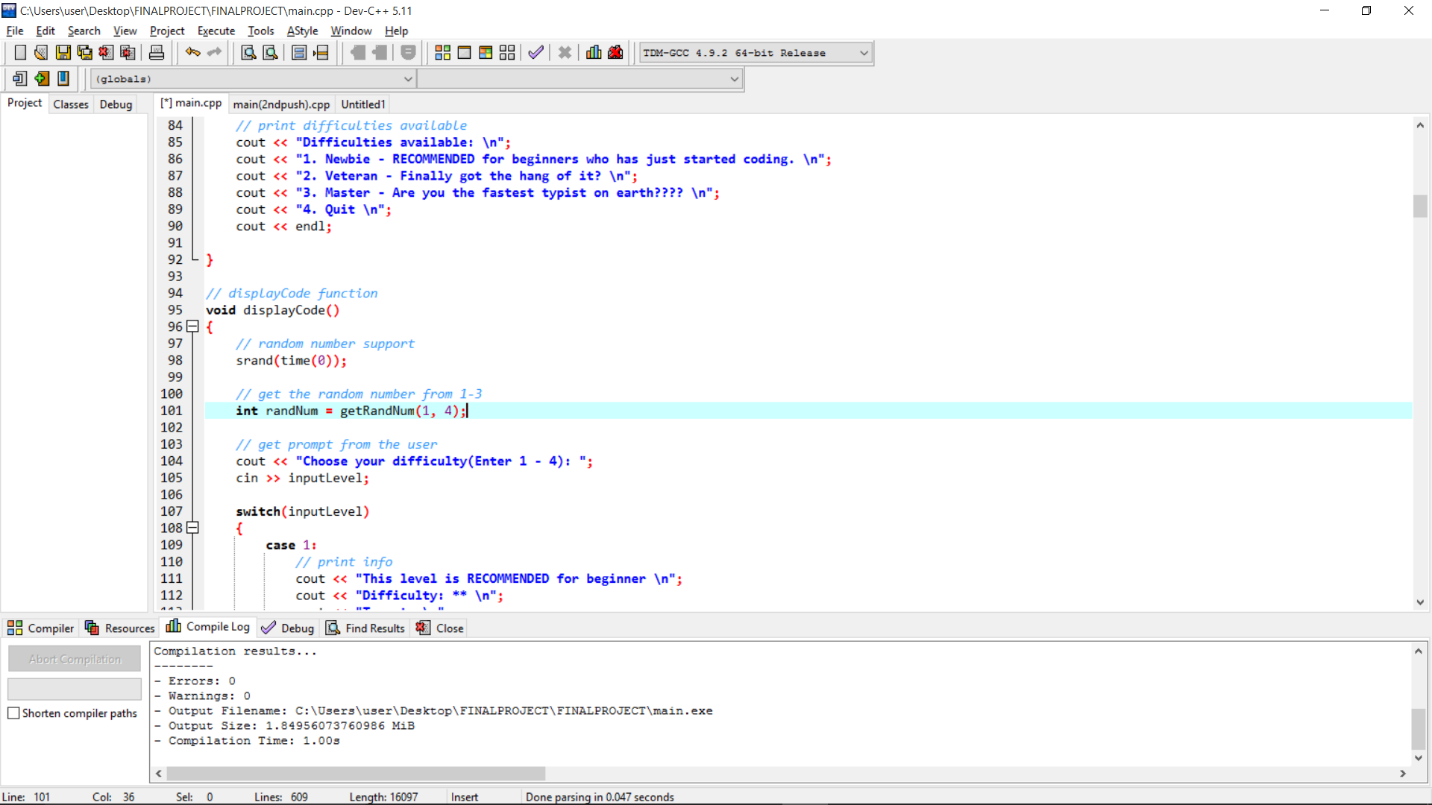
This means that there are 5 rows in that code. So, we simply just set the row limits to 5, and the loop function will be terminated after the user has input the enter key 5 times, but how about longer code? It would be a disturbing thing, because we must count the rows one by one each code, in addition, 3 codes for each level. It would be a pain isn’t it? And how about for checking the accuracy? It would be inefficient, because we need to make another for loop function again for checking each characters. Furthermore it also involves *ASCII* codes for each alphabets, including special characters such as \* (asterisks), & (ampersands), ; (semicolons), enters/spaces (\n characters) and many more. So, that’s why I revised the method and changed it.



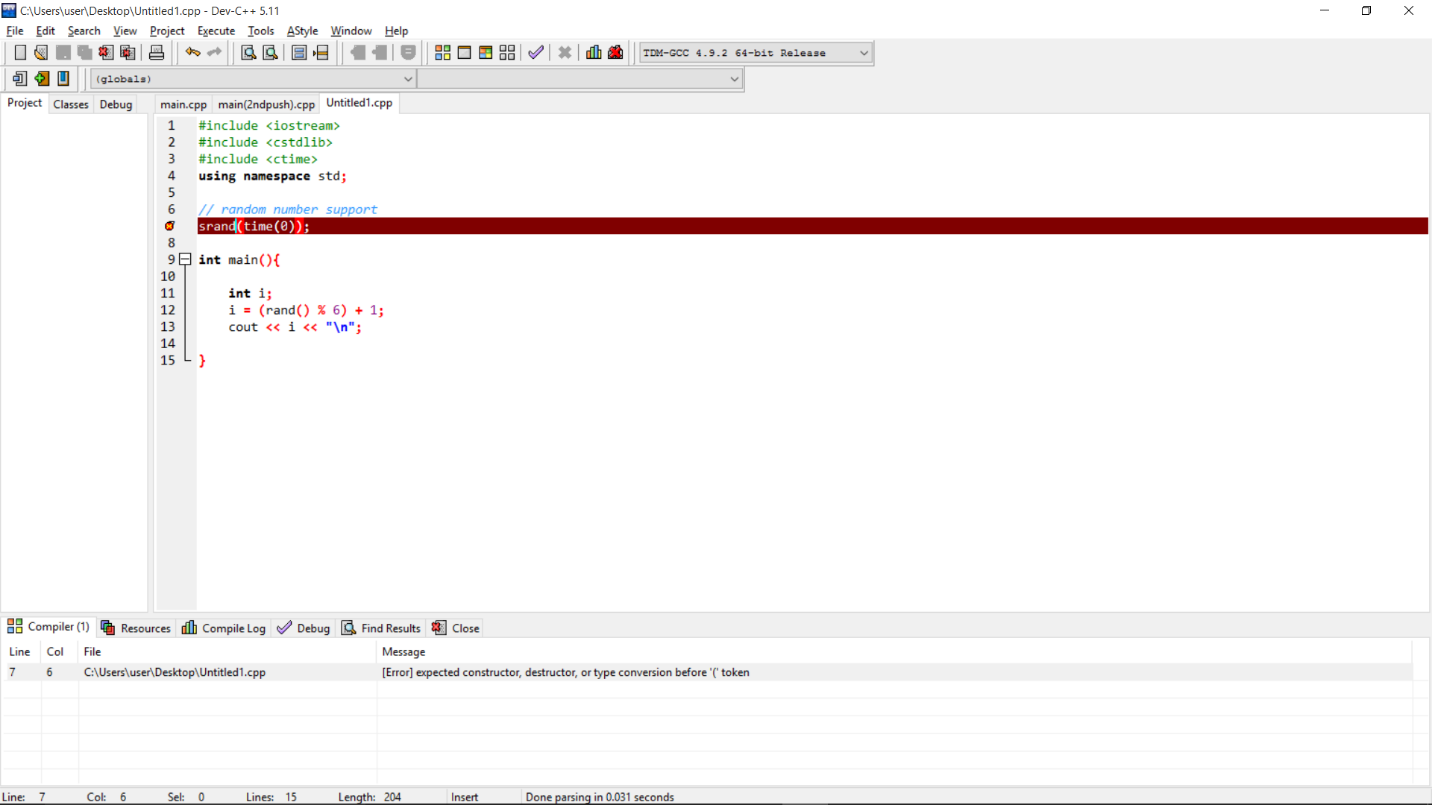
The revised version was the picture above. This method is much more efficient than the previous one. Now, the for loop function is completely removed and use the fstream function instead. I’ve created a file with .txt extension with notepad. The .txt file is holding a sample text code for each level. Then, the program will read that file with keyword ‘ifstream’. The keyword ‘cin’ followed by array of characters is for the user input (typing the code). Along with the user input, this while block will also check for errors in the typing. Inside that, we could find an if statement, indicating that if the characters which were input by the user is the same with the characters in the .txt file, then the number of correct typing is incremented by 1 for each correct typing, and the same logic applies with the number of mistypings.

* + 1. **Randomizing/Encrypting a random number**

We have discussed about how the randomized sample texts of code displayed to the screen. Using the library such as ctime and cstdlib will make it possible. Cstdlib library provides srand function. This srand initialize random number generator, the pseudo-random number generator is initialized using the argument passed as *seed*. The parameters accept an int value.



The code above is working. However, the code will not work if we put the srand outside a function. We could clearly see that srand is put inside the displayCode function without passing an arguments. The question is, what if we put srand outside a function?



This is what we got if we put the srand outside a function. ‘expected constructor, destructor, or type conversion before ‘(‘ token’. I got the similar mistake before I move the srand into a function.

CHAPTER IV

CODING SECTION

* 1. **Standard Libraries, Header, Global Variables and Function Prototypes**

// FINAL PROJECT

// --CODE TYPE EXERCISE--

// Created by: Archel Taneka

// ID: 2001585556

#include <iostream>

#include <iomanip>

#include <fstream>

#include <string>

#include <ctime>

#include <cstdlib>

#include "Account.h"

using namespace std;

// global variables

const int NUM\_CHAR = 1000;

char charArr[NUM\_CHAR]; // character array

char answer; // for input Y or N

int inputLevel; // for choosing the level

clock\_t start; // timer variable

double duration; // difference between 2 timer

ifstream inputFile; // for reading the answer key in .txt file

char ch; // user input

string fileName;

struct results{

double wpm; // calculating words per minute

int words; // count how many words from the sample text

double correctNum; // number of correct typings

double incorrectNum; // number of mistypings

double accuracy; // accuracy display in percentage

};

// declares Account class and variables support

Account nickNames;

string inputNick;

// declares structure with variable

results result;

// declares function prototypes

void getUserLevel();

void displayCode();

double getTypeUser(double);

void getResult();

int getRandNum(int, int);

void tryAgain();

* 1. **Main Function**

int main(){

// make the console program in full screen

system("mode 650");

// print welcome message and tell user what is the purpose of the program

cout << "\t-------------------------------- \n";

cout << "\t------CODE-TYPE EXERCISE-------- \n";

cout << "\t-------------------------------- \n";

cout << endl;

cout << endl;

cout << "Welcome to CODE-TYPE EXERCISE! \n";

cout << "We train you to be the fastest coder on the planet! \n";

cout << "Exercise focus: Speed & Accuracy \n";

cout << endl;

cout << endl;

// input nickname for the user

cout << "Enter nickname(alphanumeric, no spaces): ";

cin >> inputNick;

nickNames.setNickName(inputNick);

// calls the function

getUserLevel();

displayCode();

tryAgain();

return 0;

}

* 1. **getUserLevel Function**

// getUserLevel function

void getUserLevel()

{

// print difficulties available

cout << "Difficulties available: \n";

cout << "1. Newbie - RECOMMENDED for beginners who has just started coding. \n";

cout << "2. Veteran - Finally got the hang of it? \n";

cout << "3. Master - Are you the fastest typist on earth???? \n";

cout << "4. Typing God - One more.. Dare to take the challenge?? \n";

cout << "5. Quit \n";

cout << endl;

}

* 1. **displayCode Function**

// displayCode function

void displayCode()

{

// random number support

srand(time(0));

// get the random number from 1-3

int randNum = getRandNum(1, 4);

// get prompt from the user

cout << "Choose your difficulty(Enter 1 - 4): ";

cin >> inputLevel;

switch(inputLevel)

{

case 1:

// print info

cout << "This level is RECOMMENDED for beginner \n";

cout << "Difficulty: \*\* \n";

cout << "Target: \n";

cout << "Time to finish the code: 40 seconds \n";

cout << "WPM(Words Per Minute) minimum: 3 WPM \n";

cout << endl;

// ask the user to input 'Y' answer to continue

cout << "Are you ready? (Press Y to continue) \n";

cin >> answer;

cout << endl;

cout << "---------------------------------------- \n";

if(answer == 'Y' || answer == 'y')

{

// display the sample text code based on number randomed

if(randNum == 1)

{

fileName = "EasyAnswer1.txt"; // file name

result.words = 268; // set the result.word

cout << "#include <iostream> \n"

<< "using namespace std; \n"

<< endl

<< "int main(){ \n"

<< endl

<< "int firstNum; \n"

<< "int secondNum; \n"

<< endl

<< "cout << \"Enter your first number: \";\n"

<< "cin >> firstNum; \n"

<< "cout << \"Enter your second number: \";\n"

<< "cin >> secondNum; \n"

<< endl

<< "if(firstNum == secondNum){ \n"

<< "cout << \"Both numbers are identical\";\n"

<< "}else{\n"

<< "cout << \"Both numbers are not identical\"; \n"

<< "} \n"

<< endl

<< "return 0;\n"

<< "}"

<< "-------------------------------------------------- \n"

<< endl

<< "Type Now!! \n"

<< endl;

// calls the getTypeUser, getResult and tryAgain function

getTypeUser(result.words);

getResult();

tryAgain();

} else if(randNum == 2)

{

fileName = "EasyAnswer2.txt";

result.words = 177;

cout << "#include <iostream> \n"

<< "using namespace std; \n"

<< endl

<< "int main(){ \n"

<< endl

<< "string name; \n"

<< "int age; \n"

<< endl

<< "cout << \"Input name: \";\n"

<< "cin >> name; \n"

<< "cout << \"Input age: \";\n"

<< "cin >> age; \n"

<< endl

<< "cout << \"Your name is: \" << name; \n"

<< "cout << \"Your age: \" << age; \n"

<< "return 0;\n"

<< "}"

<< endl

<< "-------------------------------------------------- \n"

<< "Type Now!! \n"

<< endl;

// calls the getTypeUser and getResult function

getTypeUser(result.words);

getResult();

} else if(randNum == 3)

{

fileName = "EasyAnswer3.txt";

result.words = 189;

cout << "#include <iostream> \n"

<< "using namespace std; \n"

<< endl

<< "int main(){ \n"

<< "double payAmount = 2200.0; \n"

<< "double payPeriods = 26; \n"

<< endl

<< "double annualPay = payAmount \* payPeriods; \n"

<< endl

<< "cout << \"The total annual pay for the employee is: \" << annualPay;\n"

<< endl

<< "return 0;\n"

<< "}"

<< endl

<< "-------------------------------------------------- \n"

<< "Type Now!! \n"

<< endl;

// calls the getTypeUser, getResult and tryAgain function

getTypeUser(result.words);

getResult();

tryAgain();

}

}

break;

case 2:

// print info

cout << "This level is for intermediate typing level \n";

cout << "Difficulty: \*\*\* \n";

cout << "Target: \n";

cout << "Time to finish the code: 50 seconds \n";

cout << "WPM(Words Per Minute) minimum: 3 WPM \n";

cout << endl;

// ask the user to input 'Y' answer to continue

cout << "Are you ready? (Press Y to continue) \n";

cin >> answer;

cout << endl;

cout << "---------------------------------------- \n";

if(answer == 'Y' || answer == 'y')

{

if(randNum == 1)

{

fileName = "MediumAnswer1.txt"; // file name

result.words = 219; // set the result.word

cout << "#include <iostream> \n"

<< "using namespace std; \n"

<< endl

<< "int main(){ \n"

<< endl

<< "int num1; \n"

<< "int num2; \n"

<< "int total; \n"

<< endl

<< "cout << \"input the first number: \";\n"

<< "cin >> num1; \n"

<< "cout << \"input the second number: \";\n"

<< "cin >> num2; \n"

<< endl

<< "total = num1 + num2; \n"

<< "cout << \"The total sum of the two number is: \" << total; \n"

<< endl

<< "return 0; \n"

<< "}"

<< endl

<< "-------------------------------------------------- \n"

<< "Type Now!! \n "

<< endl;

// calls the getTypeUser, getResult and tryAgain function

getTypeUser(result.words);

getResult();

tryAgain();

} else if(randNum == 2)

{

fileName = "MediumAnswer2.txt"; // file name

result.words = 244; // set the result.word

cout << "#include <iostream> \n"

<< "using namespace std; \n"

<< endl

<< "int main(){ \n"

<< endl

<< "int number; \n"

<< endl

<< "cout << \"Enter a number: \";\n"

<< "cin >> number; \n"

<< endl

<< "if(number == 1){\n"

<< "cout << \"My subject\"; \n"

<< "}else if(number == 2){\n"

<< "cout << \"Lecturer/Facilitators\"; \n"

<< "}else if(number == 3){"

<< "cout << \"Shared Materials\"; \n"

<< "}else {\n"

<< "cout << \"Invalid number!\"; \n"

<< "}"

<< endl

<< "-------------------------------------------------- \n"

<< "Type Now!! \n "

<< endl;

// calls the getTypeUser, getResult and tryAgain function

getTypeUser(result.words);

getResult();

tryAgain();

} else if(randNum == 3)

{

fileName = "MediumAnswer3.txt"; // file name

result.words = 313; // set the result.word

cout << "#include <iostream> \n"

<< "using namespace std; \n"

<< endl

<< "int main(){ \n"

<< endl

<< "unsigned int distance; \n"

<< "unsigned int speed; \n"

<< "unsigned int time; \n"

<< "int i; \n"

<< endl

<< "cout << \"Enter a speed of a vehicle in mph: \"; \n"

<< "cin >> speed;\n"

<< "cout << \"How many hours has it traveled? \"; \n"

<< "cin >> time;\n"

<< "for(i = 0; i <= time; i++){\n"

<< "distance = speed \* i; \n"

<< "cout << i << \" hour(s)\" << " " << distance << \" distance traveled\" << endl; \n"

<< "}"

<< endl

<< "return 0; \n"

<< "}"

<< endl

<< "-------------------------------------------------- \n"

<< "Type Now!! \n "

<< endl;

// calls the getTypeUser, getResult and tryAgain function

getTypeUser(result.words);

getResult();

tryAgain();

}

}

break;

case 3:

// print info

cout << "WARNING: This level is for TYPING MASTER \n";

cout << "Difficulty: \*\*\*\*\* \n";

cout << "Target: \n";

cout << "Time to finish the code: 60 seconds \n";

cout << "WPM(Words Per Minute) minimum: 4 WPM \n";

cout << endl;

cout << endl;

cout << "---------------------------------------- \n";

cout << "Are you ready? (Press Y to continue) \n";

cin >> answer;

if(answer == 'Y' || answer == 'y')

{

if(randNum == 1)

{

fileName = "HardAnswer1.txt";

result.words = 303; // set the result.word

cout << "#include <iostream> \n"

<< "using namespace std; \n"

<< endl

<< "int main(){\n"

<< endl

<< "int columnSpaces; \n"

<< "int rowSpaces; \n"

<< "int dotsIncrement; \n"

<< "int dots = 1; \n"

<< endl

<< "for(columnSpaces = 1; columnSpaces <= 5; columnSpaces++) \n"

<< "{ \n"

<< "for(rowSpaces = 4; rowSpaces >= columnSpaces; rowSpaces--) \n"

<< "{ \n"

<< "cout << \" \"; \n"

<< "} \n"

<< "for(dotsIncrement = 0; dotsIncrement < dots; dotsIncrement++) \n"

<< "{ \n"

<< "cout << \"\*\"; \n"

<< "} \n"

<< "cout << endl; \n"

<< "dots += 2; \n"

<< "}"

<< endl

<< "-------------------------------------------------- \n"

<< "Type Now!! \n"

<< endl;

// calls the getTypeUser, getResult and tryAgain function

getTypeUser(result.words);

getResult();

tryAgain();

} else if(randNum == 2)

{

fileName = "HardAnswer2.txt";

result.words = 283; // set the result.word

cout << "#include <iostream> \n"

<< "using namespace std; \n"

<< endl

<< "int function(int array[], int arraySize, int n); \n"

<< endl

<< "int main(){\n"

<< endl

<< "const int ARRAY\_SIZE = 5; \n"

<< "int array[ARRAY\_SIZE] = {6, 7, 8, 9, 10}; \n"

<< "int n = 3; \n"

<< endl

<< "function(array, ARRAY\_SIZE, n); \n"

<< "} \n"

<< endl

<< "int function(int array[], int arraySize, int n) \n"

<< "{ \n"

<< "for(int i = 3; i < arraySize; i++) \n"

<< "{ \n"

<< "if(i > n){ \n"

<< "cout << arraySize << endl; \n"

<< "} \n"

<< "} \n"

<< "} \n"

<< "-------------------------------------------------- \n"

<< "Type Now!! \n"

<< endl;

// calls the getTypeUser, getResult and tryAgain function

getTypeUser(result.words);

getResult();

tryAgain();

} else if(randNum == 3)

{

fileName = "HardAnswer3.txt";

result.words = 499; // set the result.word

cout << "#include <iostream> \n"

<< "#include <iomanip> \n"

<< "using namespace std; \n"

<< endl

<< "double getHeight(); \n"

<< "double getRadius(); \n"

<< "double square(double); \n"

<< endl

<< "int main(){\n"

<< endl

<< "double const phi = 3.1459; \n"

<< "double radius; \n"

<< "double height; \n"

<< "double volume; \n"

<< endl

<< "radius = getRadius(); \n"

<< "height = getHeight(); \n"

<< "volume = phi \* square(radius) \* height \* 1/3; \n"

<< endl

<< "cout << \"The volume of the cone is: \" << volume; \n"

<< "return 0; \n"

<< "} \n"

<< endl

<< "double getHeight(){ \n"

<< "double height; \n"

<< "cout << \"Enter height: \"; \n"

<< "cin >> height; \n"

<< "return height; \n"

<< "} \n"

<< endl

<< "double getRadius(){ \n"

<< "double radius; \n"

<< "cout << \"Enter radius: \"; \n"

<< "cin >> radius; \n"

<< "return radius; \n"

<< "} \n"

<< endl

<< "double square(double num){ \n"

<< "return num \* num; \n"

<< "}"

<< endl

<< "-------------------------------------------------- \n"

<< "Type Now!! "

<< endl;

// calls the getTypeUser, getResult and tryAgain function

getTypeUser(result.words);

getResult();

tryAgain();

}

}

break;

case 4:

// print info

cout << "WARNING: COME BACK HERE WHEN YOU ARE SKILLED ENOUGH! \n";

cout << "Difficulty: unknown \n";

cout << "Target: \n";

cout << "Time to finish the code: 300 seconds \n";

cout << "WPM(Words Per Minute) minimum: 3 WPM \n";

cout << endl;

cout << endl;

cout << "---------------------------------------- \n";

// ask the user to input 'Y' answer to continue

cout << "Are you ready? (Press Y to continue) \n";

cin >> answer;

if(answer == 'Y' || answer == 'y')

{

fileName = "GodAnswer.txt"; // file name

result.words = 704; // set the result.word

cout << "#include <cassert> \n"

<< "#include <cstdlib> \n"

<< "#include <iostream> \n"

<< "using namespace std; \n"

<< endl

<< "class Cpu{ \n"

<< "public: \n"

<< "virtual int dummy(int, int){} \n"

<< "private: \n"

<< "virtual int add\_(int a, int b) {return a + b;} \n"

<< "virtual int sub\_(int a, int b) {return a - b;} \n"

<< "virtual int mul\_(int a, int b) {return a \* b;} \n"

<< "virtual int div\_(int a, int b) {return a / b;} \n"

<< "virtual int mod\_(int a, int b) {return a % b;} \n"

<< "virtual int and\_(int a, int b) {return a & b;} \n"

<< "virtual int or\_(int a, int b) {return a | b;} \n"

<< "virtual int xor\_(int a, int b) {return a ^ b;} \n"

<< "}; \n"

<< endl

<< "int main(){"

<< "typedef int (Cpu::\*Memfun)(int, int); \n"

<< endl

<< "union{ \n"

<< "Memfun fn; \n"

<< "unsigned char ptr[6]; \n"

<< "}u; \n"

<< endl

<< "Cpu cpu; \n"

<< "u.fn = &Cpu::dummy; \n"

<< "assert(argc == 4); \n"

<< endl

<< "int p1 = atoi(argv[1]); \n"

<< "int p2 = atoi(argv[3]); \n"

<< "char op = argv[2][0]; \n"

<< endl

<< "assert(op >= 'A' && op <= 'H'); \n"

<< "u.ptr[0] = 1 + 4 \* (op - 'A' + 1); \n"

<< endl

<< "cout << \"The answer is \" << ((cpu.\*(u.fn))(p1, p2)) << endl; \n"

<< endl

<< "return 0; \n"

<< "}"

<< endl

<< "---------------------------------------------------------- \n"

<< "Type Now!! "

<< endl;

// calls the getTypeUser, getResult and tryAgain function

getTypeUser(result.words);

getResult();

tryAgain();

}

case 5:

cout << "See you again! \n";

cout << endl;

cout << "Keep \n"

<< "Calm \n"

<< "and \n"

<< "Keep \n"

<< "Coding"

<< "Cheers! :D";

exit(EXIT\_FAILURE); // terminate the program

default:

cout << "Invalid input! Please enter number between 1-4! \n";

displayCode(); // calls the function itself again

}

}

* 1. **getTypeUser Function**

// getTypeUser function

double getTypeUser(double words)

{

int counter; // array counter

// start the timer

start = clock();

// open Answer1.txt file

inputFile.open(fileName);

// get user input and also check for mistypings

while(inputFile >> ch)

{

cin >> charArr[counter];

if(ch == charArr[counter])

{

result.correctNum += 1;

} else {

result.incorrectNum += 1;

}

counter ++;

}

// close Answer1.txt file

inputFile.close();

// stop the timer, calculate wpm, accuracy and display message

cout << endl;

// display nickname

cout << nickNames.getNickName() << "'s stats:" << endl;

cout << endl;

duration = (clock() - start) / (double)CLOCKS\_PER\_SEC;

cout << "You finished the code in: " << duration << " s" << endl;

result.wpm = words / duration;

cout << "You typed: " << result.correctNum << " correct characters" << endl;

cout << "You typed: " << result.incorrectNum << " incorrect characters" << endl;

result.accuracy = result.correctNum / result.words \* 100;

// set output format

cout << setprecision(2) << fixed << showpoint;

cout << "Accuracy: " << result.accuracy << "%" << endl;

cout << "Your typing speed: " << result.wpm << " wpm" << endl;

return words;

}

* 1. **getResult Function**

// getResult function

void getResult()

{

// check for time and wpm

if(inputLevel == 1)

{

if(duration > 40 && result.wpm < 2)

{

cout << "Too slow! Practice more! Your target is < 50 seconds with 3 WPM!";

cout << "It's okayy.. Practice more! n" << endl;

} else {

cout << "Great job! Try the next level! \n";

}

} else if(inputLevel == 2)

{

if(duration > 50 && result.wpm < 3)

{

cout << "Too slow! Practice more! Your target is <= 40 seconds with 3 WPM!";

cout << "Beat the easy level first \n!" << endl;

} else {

cout << "Great job! Try the next level! \n";

}

} else if(inputLevel == 3)

{

if(duration > 60 && result.wpm < 4)

{

cout << "Too slow! Practice more! Your target is <= 100 seconds with 4 WPM!" << endl;

cout << "Beat the medium level first \n!" << endl;

} else {

cout << "Well Done! You've conquered the master level!! Now, you are the a code typing master!";

}

} else if(inputLevel == 4)

{

if(duration > 60 && result.wpm < 4)

{

cout << "Practice before taking this level!! Your target is <= 300 seconds with 3 WPM" << endl;

cout << "Beat the master level first \n!" << endl;

} else {

cout << "You did it!! You’re now the typing GOD!!";

}

}

}

* 1. **getRandNum Function**

// getRandNum function

int getRandNum(int low, int high)

{

return rand() % (high - low) + low;

}

* 1. **tryAgain Function**

// tryAgain function

void tryAgain()

{

// asks the user to play again

char answer;

cout << "Try again? (Y/N): ";

cin >> answer;

if(answer == 'Y' || answer == 'y')

{

system("CLS"); // clears everything on display output

main(); // calls the main function

} else if(answer == 'N' || answer == 'n')

{

cout << "Thank you for playing! See you again!";

exit(EXIT\_FAILURE);

} else {

cout << "Invalid command!";

tryAgain(); // call the function itself again

}

}

CHAPTER V

LOGBOOK & DOCUMENTATION

* 1. **October 6th 2016(First Commit)**

Created a new repository and added a flowchart.

* 1. **October 7th 2016**

Added WPM feature and description of the program to the flowchart.

* 1. **October 11th 2016**

Edited the description of the program

* 1. **October 12th 2016**

Created main.cpp then added the choosing menu interface and welcome message.

* 1. **October 20th 2016(Second Commit)**
* Edited the flowchart
* More levels added (easy, medium, hard), timer, wpm and user input (typing the code) feature to the main.cpp
  1. **October 27th 2016(Third Commit)**
* Added .txt files for ifstream (Answer1.txt, Answer2.txt and Answer3.txt)
* Added the accuracy, number of correct and mistypings feature.
  1. **October 29th 2016**

Added a random number generator feature for a random sample text code.

* 1. **November 3rd 2016**

Added more sample texts code in easy and medium level.

* 1. **November 4tj 2016**
* Added more sample texts code in hard level
* Added input nickname feature
  1. **November 5th 2016(Last Commit)**
* Added Level: GOD
* Fix the minimum standard of timer and typing speed in each level.

CHAPTER V

CONCLUSION & FEEDBACK

The project that I made is actually one of a simple program that we can make with C++ programming language. There are more possibilities that we can make even more complex and useful programs for our daily life. Any suggestions and feedbacks are welcomed in order to fix and develop the program at its best. Author expresses his gratitude, especially towards facilitators and lecturers: Mr. Ida Bagus Kerthyayana Manuaba, and Mr. Stavin Deeswe also my fellow colleague, Aldi Radityo for making this project possible.

Last but not least, finally author say thank you for everyone who has spared time reading this final project report, and I am sorry if there were any mistakes or errors written in the report.