CODE-TYPE EXERCISE

CREATED BY : ARCHEL TANEKA

ID : 2001585556

Program Description

As a computer science students, typing a code is like our daily exercise. We keep practicing typing the code in order to get something, whether it is just a regular practice or to understand more on what you’ve learnt today on class. But that’s not the case, you’re just expand your knowledge and understanding about the materials. Imagine this, you’re in a class and it will finish in 15 minutes, but suddenly your lecturer/facilitator gives you an individual assignment/task. Your task is to make a simple program and you know that it might be 200+ lines of codes or worse case, more than 200 lines. It’s a real pain isn’t it?

So, here’s the purpose of the program. This program will help you practice your typing skills, especially code-typing skills in C++, because we already knew that typing a program is different from typing another assignments such as typing essays or reports. Typing a bunch of codes includes many of symbols like #(hashtags), \*(asterisks), &(ampersand), ;(semicolon) and many more. You will choose from different levels (beginner, medium or hard). Then the user types the code exactly the same with the example code. At the end, the program will calculate your wpm (words per minute) and time that was required to finish the code. Then it will tell you whether you’re fast enough to type that code, or too slow to finish the code.

Objective : Type as fast as you can with minimal errors (accuracy) and as similar as the example code if possible.

Function(s):

1. Main Function

The entire program are written here, including the function callings.

1. getUserLevel

This function will get prompt from the user to choose the level provided, whether it is easy, medium or hard. After the user chose the level, a sample text code will be shown according to a random number generator.

1. getTypeUser

This is the function where user has to type the code similar with the sample text code. Furthermore, it’s also check for mistyping and correct typing. The addition of ifstream library will read input file from .txt extension and check the number of correct and wrong characters. It will be increased by 1 for each mistyping and correct typing and displays WPM (Words Per Minute), timer when the user finished and accuracy in percentage.

1. displayCode

This function consists of the sample texts of code, the timer is also turned on here. When the user has done typing, the timer also stops

1. getResult

When the user finished typing the sample code, this function will check the results. It will check the user’s typing speed/WPM, timer and accuracy correspond to the level, whether it’s easy, medium or hard. Every level, has its own standard of WPM and timer. If the results meet the minimum requirements, a congratulatory message will be displayed, if not then another message will be displayed.

1. getRandNum

This function will generate a random number generator. There are 3 different levels with another 3 variations, so the user won’t get bored with the same sample text code. If the random number generator generates number 1, then 3 possible sample texts of code will be displayed to the screen. Same rule applies to number 2 and 3.

1. tryAgain

This function will simply ask the user at the end of the program whether they want to try again or not. If the user is willing to try again, the main function will be called, if not then the program terminates.

Class(es):

1. Account

This class simply stores a nickname of the user. It then will be displayed at the end of the program along with the stats, such as typing speed, accuracy, timer and numbers of correct and wrong typings.

**FLOWCHART**

START

Print welcome message

Input Nickname & Level

Easy

Easy?

Hard?

Easy sample codes

Hard

Hard sample codes

Random number generator

Display random text code based on number encrypted

Timer ON

Check for accuracy

User input: type the sample code

Displays Nickname, WPM, timer, correct & wrong characters and accruacy

Correct typing?

Mistyping?

correct

Correct counter +1

Timer OFF

wrong

Wrong counter +1

Displays accuracy

Calculates accuracy

Timer > x?  
WPM < y?

Timer < x?  
WPM > y?

Displays try again message

<x

>y

>x

<y

Displays congratulatory message

Yes

Try again?

No

END