A green logo with a snake and text

Description automatically generated

เอกสารประกอบการใช้งานโปรแกรม

กลุ่ม JavaScript Mastery

โดย

นายวรินทร์ สายปัญญา รหัสนักศึกษา 6630250435 หมู่ 881

อาจารย์ผู้สอน

อาจารย์วนิดา คำประไพ

อาจารย์จิรวรรณ เจริญสุข

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา 01418113 การโปรแกรมคอมพิวเตอร์

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ ศรีราชา มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา

ประจำภาคปลาย ปีการศึกษา 2566

**สารบัญ**

หน้า

Source code ของโปรเจค 3 - 33

เกี่ยวกับโปรแกรม 34

ฟีเจอร์ของโปรแกรม 35

โครงสร้างของโปรเจค 36

อธิบายผลลัพธ์และสิ่งที่ต้องป้อนเข้าไปในโปรแกรม 37 - 71

แหล่งเอกสารอ้างอิงและการใช้งานคำสั่ง APIs ทั้งหมดของ Libraries 72

**Source code ของโปรเจค**

โปรเจคนี้ถูกทำขึ้นมาเพื่อส่งเป็นโปรเจคในรายวิชา การโปรแกรมคอมพิวเตอร์(Computer Programming) โดยโค้ดโปรแกรมนี้ได้อัปขึ้นผ่านบนเว็บไซต์ Github.com เพื่อง่ายต่อการแชร์โปรเจคแล้วง่ายต่อการเข้าถึง



สามารถเปิดตาม url นี้ได้เลย <https://github.com/WarinCode/CPP-project>

Short url : https://bit.ly/3VgBKNp

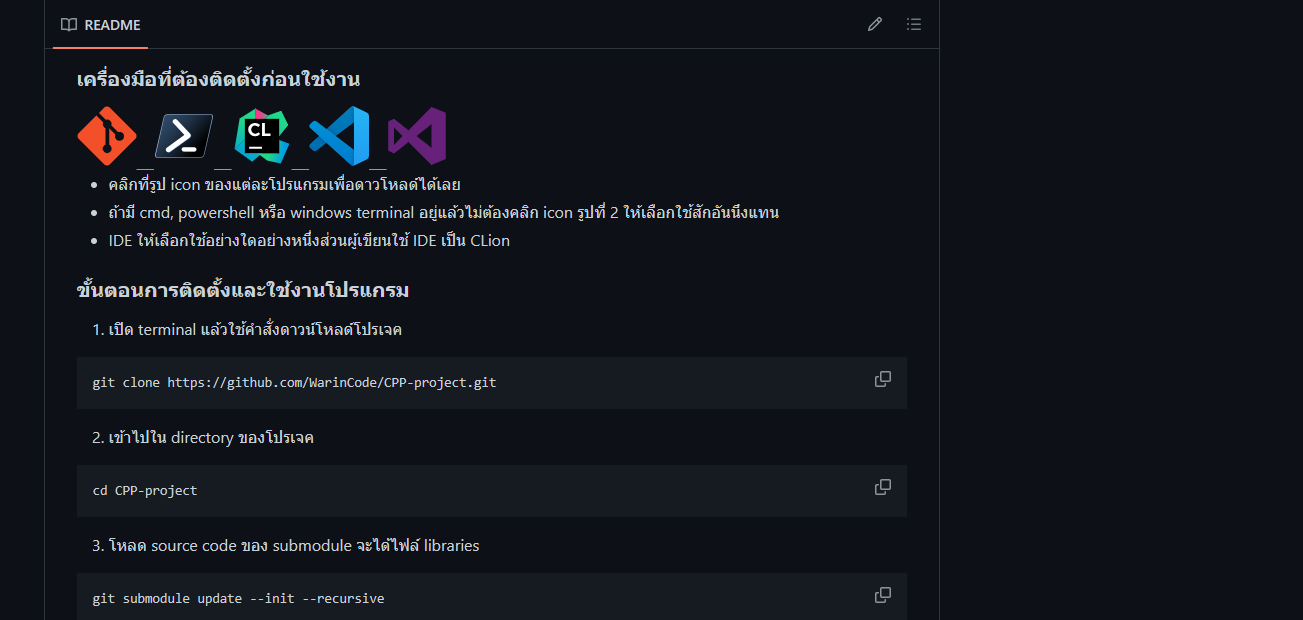
หรือ Scan QR-code นี้



*/\* Documents  
 \* https://marcuscode.com/lang/cpp/files  
 \* https://medium.com/@vachirachat.saw/%E0%B9%82%E0%B8%84%E0%B8%A3%E0%B8%87%E0%B8%AA%E0%B8%A3%E0%B9%89%E0%B8%B2%E0%B8%87%E0%B8%82%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%B9%E0%B8%A5%E0%B8%95%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%86%E0%B9%83%E0%B8%99-c-1ccdb706c38e  
 \* https://www.devdit.com/post/4200/c-plus-plus-vector-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B8%97%E0%B8%B3%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%A2%E0%B8%B1%E0%B8%87%E0%B9%84%E0%B8%87#gsc.tab=0  
 \* https://medium.com/@marktbss/c-hackerrank-vector-erase-11c65b830a43  
 \* https://www.geeksforgeeks.org/how-to-clear-console-in-cpp  
 \* https://www.javatpoint.com/cpp-date-and-time  
 \* https://github.com/seleznevae/libfort  
 \* https://seleznevae.github.io/libfort/index.html  
 \* https://cplusplus.com/reference/cstdlib/rand  
 \* https://github.com/ikalnytskyi/termcolor  
 \* https://termcolor.readthedocs.io/#  
 \* https://weerasak.dev/posts/2023/03/18/basic-cmake-for-building-c-cpp-project/  
 \*/  
  
/\* รายชื่อสมาชิกในกลุ่มที่เขียนโปรแกรมนี้  
 \* 1. นาย วรินทร์ สายปัญญา รหัสนิสิต 6630250435 หมู่เรียน ภาคปฎิบัติ 881  
\*/  
  
/\* ข้อบังคับการใช้งานโปรแกรม  
 \* 1. การตั้งชื่อสินค้าต้องตั้งชื่อที่ติดกันไม้เว้นวรรคเพราะหากตั้งชื่อที่มีความยาวและก็เว้นวรรคชื่อจะทำให้การอ่านไฟล์ข้อมูลผิดพลาดหากจะจำเป็นต้องตั้งชื่อยาวให้ใช้ - หรือ \_ ขั้นแต่ละคำไว้  
 \* 2. การแก้ไขไฟล์ data.txt มีผลโดยตรงต่อตัวโปรแกรมเพราะฉะนั้นห้ามแก้ไขไฟล์ data.txt เด็ดขาด  
 \* 3. ห้ามย้ายไฟล์ หรือ ลบโฟลเดอร์ txts เพราะจะมีผลกับการอ่านเขียนข้อมูลในตัวโปรแกรม หัามลบโฟลเดอร์ txts และ ไฟล์ data.txt เด็ดขาด  
 \* 4. หากมีผู้ใดสนใจจะนำโปรแกรมนี้ไปพัฒนาต่อยอดสามารถเชิญนำไปพัฒนาต่อได้เลย  
 \* 5. ห้าม คัดลอก (copy code) ไฟล์โปรเจค main.cpp ไปส่งเป็นงานโปรเจคคของกลุ่มตัวเองเป็นเด็ดขาด ถ้าหากจับว่าจับได้ จะไปฟ้องอาจารย์  
 \*\*/  
  
#include* <iostream>  
*#include* <string>  
*#include* <fstream>  
*#include* <sstream>  
*#include* <vector>  
*#include* <ctime>  
*#include* <cstdlib>  
*#include* <fort.hpp>  
*#include* <fort.h>  
*#include* <termcolor/termcolor.hpp>  
  
*using namespace* std;  
*using namespace* fort;  
*using namespace* termcolor;  
  
*// กำหนดจำนวนสินค้าตอนเริ่มต้น มี 20 จำนวน ของแต่ละสินค้า  
#define* STOCK 20  
*#define* NUMBER\_CATEGORIES 17  
*// ประเภท หรือ หมวดหมู่สินค้า  
const* string productCategories[NUMBER\_CATEGORIES] = { "phone", "tablet", "laptop", "computer", "car",  
 "health and beauty","game", "bag", "electrical appliance", "pet",  
 "camera", "shoes", "watch", "sport", "musical instrument",  
 "furniture", "food"  
};  
  
*// namespace ของโปรแกรมไว้เก็บ functions ที่ไว้ใช้งาน  
namespace* program{  
 *// ประกาศ function prototypes ไว้ล่วงหน้า  
 void* showOptions();  
 *void* showErrorMessage(string message);  
 *void* showErrorMessage(*int* size, string message[]);  
 *void* showSuccessfulMessage(string message);  
 *void* showProductCategories();  
 *int* generateId(*int* from, *int* to);  
 string addZeroNumber(*int* num);  
 *bool* yesOrNo(string yn);  
}  
  
*// class Time สำหรับการใช้บอกวันเวลาปัจจุบัน  
class* Time{  
*public*:  
 time\_t now; *// เวลาปัจจุบัน* string dt; *// datetime* tm\* ltm; *// localtime ต้องใช้เป็น pointer  
  
 // attributes วันที่และเวลา  
 int* year;  
 *int* month;  
 *int* weekday;  
 *int* day;  
 *int* hours;  
 *int* minutes;  
 *int* seconds;  
  
 *// array วัน และ เดือน* string days[7] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" };  
 string months[12] = {"January", "Febuary", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December" };  
  
 *// constructor method* Time(){  
 *// เวลาปัจจุบัน* now = time(0);  
 *// ส่ง address now เข้าไปใน function ctime และ localtime* dt = ctime(&now);  
 ltm = localtime(&now);  
 *// เข้าถึงตัวแปรข้างในของ struct tm* hours = ltm -> tm\_hour;  
 minutes = ltm -> tm\_min;  
 seconds = ltm -> tm\_sec;  
 year = ltm -> tm\_year + 1900; *// ต้องบวก 1900 ไปด้วยถึงจะเป็นปีล่าสุด  
 // attribute month และ weekday ใช้คู่กับ array ได้เพราะสามารถใช้เลขเป็นเลข index ของ array* month = ltm -> tm\_mon; *// เลขระหว่าง 0 - 11* weekday = ltm -> tm\_wday; *// เลขระหว่าง 0 - 6* day = ltm -> tm\_mday; *// เลขระหว่าง 1 - 31* }  
  
 *// getter methods* string getDate(){  
 *return* dt;  
 }  
 *int* getYear(){  
 *return* year;  
 }  
 *int* getMonth(){  
 *// ต้องบวก 1 เพราะ tm\_mon คืนเลขกลับมาเป็น 0 - 11  
 return* month + 1;  
 }  
 *int* getDay(){  
 *return* day;  
 }  
 *int* getHours(){  
 *return* hours;  
 }  
 *int* getMinutes(){  
 *return* minutes;  
 }  
 *int* getSeconds(){  
 *return* seconds;  
 }  
 *int* getWeekDay(){  
 *return* weekday;  
 }  
 *// คืนกลับมาเป็นข้อความ สมาชิกใน array* string getDays(){  
 *return* days[weekday];  
 }  
 string getMonths(){  
 *// ลบเลขออกไป 1 ถึงจะใช้เลข index 0 - 11  
 return* months[getMonth() - 1];  
 }  
};  
  
*// Superclass class Product เป็น class ต้นแบบที่ให้ subclass สืบทอดคุถสมบัติและพฤติกรรมต่างๆของคลาสนี้  
class* Product {  
*public*:  
 *// attributes  
 int* id; *// รหัสสินค้า* string name; *// ชื่อสินค้า  
 float* price; *// ราคาสินค้า  
 int* stock; *// จำนวนสินค้าที่เก็บไว้ในคลัง* string category; *// ประเภทของสินค้า* string brand; *// แบรนด์ของสินค้า  
 // constructor method* Product(*int* Id, string Name, *float* Price, *int* Stock = STOCK, string Brand = "-", string Category = "-"){  
 *// เมื่อสร้าง object ให้รับค่า arguments ที่ส่งมาจาก constructor แล้วมาเก็บไว้ใน attributes* setId(Id);  
 setName(Name);  
 setPrice(Price);  
 setStock(Stock);  
 setCategory(Category);  
 setBrand(Brand);  
 }  
  
 *// constructor method (overloading) สำหรับการสร้าง object ให้มีค่าเริ่มต้น* Product(string Category = "-"){  
 setId(0);  
 setName("-");  
 setPrice(0);  
 setStock(STOCK);  
 setCategory(Category);  
 setBrand("-");  
 }  
  
 *// getter methods ให้ข้อมูลใน attribute  
 int* getId(){  
 *return* id;  
 }  
 string getName(){  
 *return* name;  
 }  
 *float* getPrice(){  
 *return* price;  
 }  
 *int* getStock(){  
 *return* stock;  
 }  
 string getCategory(){  
 *return* category;  
 }  
 string getBrand(){  
 *return* brand;  
 }  
  
 *// setter methods แก้ไขข้อมูลใน attribute  
 void* setId(*int* Id){  
 id = Id;  
 }  
 *void* setName(string Name){  
 name = Name;  
 }  
 *void* setPrice(*float* Price){  
 price = Price;  
 }  
 *void* setStock(*int* Stock){  
 stock = Stock;  
 }  
 *void* setCategory(string Category){  
 category = Category;  
 }  
 *void* setBrand(string Brand){  
 brand = Brand;  
 }  
};  
  
*// สร้างตัวแปร data เก็บข้อมูลสินค้าทั้งหมดจากในไฟล์ data.txt และ ข้อมูลที่ เพิ่ม , ลบ และ แก้ไขเข้ามา*vector<Product> data = {};  
  
*// Subclass class Order ใช้สำหรับจัดเก็บข้อมูลสินค้าที่ผู้ใช้งานสั่งเข้ามา  
class* Order: *public* Product {  
*public*:  
 *int* quantity; *// จำนวนสินค้า  
 float* sum; *// ยอดรวมของสินค้านั้น  
  
 // ส่งค่า parameters ไปให้ constructor ใน class Product จัดการ* Order(*int* Id, string Name, *float* Price, string Brand = "-", string Category = "-", *int* Stock = STOCK): Product(Id, Name, Price, Stock, Brand, Category){}  
  
 *// getter methods  
 int* getQuantity(){  
 *return* quantity;  
 }  
 *float* getSum(){  
 *return* sum;  
 }  
  
 *// setter methods  
 void* setQuantity(*int* Quantity){  
 quantity = Quantity;  
 }  
 *void* setSum(*float* Sum){  
 sum = Sum;  
 }  
};  
  
*// class ReceiveProduct ใช้สำหรับการรับข้อมูลสินค้าที่อ่านได้จากไฟล์ข้อมูล data.txt และ ไว้จัดการเก็บข้อมูลเป็นกลุ่มเมื่อ loop ข้อมูลมาจากตัวแปร data ได้  
class* ReceiveProduct {  
*public*:  
 string name;  
 *int* id;  
 *float* price;  
 *int* stock;  
 string category;  
 string brand;  
};  
  
*// class File ใช้ในการจัดการไฟล์ data.txt และ orders.txt เพื่อเขียนและอ่านข้อมูล  
class* File {  
*public*:  
 *// method อ่านข้อมูลในไฟล์ data.txt แล้วมาเก็บไว้ในตัวแปร data  
 static void* read(string path = *R*"(C:\Users\ACER USER5949486\Desktop\CPP-project\txts\data.txt)", *bool* showMessage = *false*){  
 *// ตัวแปรสำหรับอ่านไฟล์ข้อมูล* ifstream readFile;  
 *// เปิดไฟล์เพื่ออ่านข้อมูล* readFile.open(path ,ios::in);  
 *// เช็คว่าสามารถเเปิดไฟล์ได้หรือไม่  
 if*(readFile.is\_open()){  
 *// ล้างข้อมูลทั้งหมดที่เก็บไว้ในตัวแปร data* ::data.clear();  
 string line;  
 *// loop อ่านไฟล์ data.txt ทีละบรรทัด  
 while*(getline(readFile, line)){  
 *// สร้าง object สินค้าเพื่อมารอรับข้อมูลสินค้าที่อ่านได้* ReceiveProduct rp;  
 *// สร้างตัวแปร string stream สำหรับเก็บข้อความทีละบรรทัด* stringstream ss(line);  
 *// ให้ตัวแปร ss นำเข้าข้อมูลสินค้าทีละตัวแปร  
 // ในไฟล์ data.txt จะอ่านข้อมูลตามนี้ในแต่ละบรรทัด: id name price stock brand category* ss >> rp.id >> rp.name >> rp.price >> rp.stock >> rp.brand >> rp.category;  
 *// สร้าง object เพื่อเก็บข้อมูลสินค้าเข้าตัวแปร data* Product product = Product(rp.id, rp.name, rp.price, rp.stock, rp.brand, rp.category);  
 *// เก็บข้อมูลทีละ object* ::data.push\_back(product);  
 }  
 *if*(showMessage) program::showSuccessfulMessage("Read file completed.");  
 } *else* {  
 *if*(showMessage) program::showErrorMessage("Cannot open file data.txt to read data!");  
 }  
 readFile.close();  
 };  
  
 *// method เขียนข้อมูลลงในไฟล์ data.txt ด้วยข้อมูล vector Product  
 static void* write(string path = *R*"(C:\Users\ACER USER5949486\Desktop\CPP-project\txts\data.txt)", *bool* showMessage = *false*){  
 *// ตัวแปรสำหรับเขียนไฟล์ข้อมูล* ofstream writeFile;  
 *// เปิดไฟล์เพื่อเขียนข้อมูล* writeFile.open(path, ios::out);  
 *// เช็คว่าสามารถเเปิดไฟล์ได้หรือไม่  
 if*(writeFile.is\_open()){  
 *// loop ข้อมูลตัวแปร data  
 for*(Product item : ::data){  
 *// เขียนข้อมูลทีละบรรทัด โดยข้อมูลสินค้าแต่ละส่วนจะเว้นระยะห่าง 1 tab* writeFile << item.getId() << "\t" << item.getName() << "\t" << item.getPrice() << "\t" << item.getStock() << "\t" << item.getBrand() << "\t" << item.getCategory() << endl;  
 }  
 *if*(showMessage) program::showSuccessfulMessage("Write file completed.");  
 } *else* {  
 *if*(showMessage) program::showErrorMessage("Cannot open file data.txt to write data!");  
 }  
 writeFile.close();  
 }  
  
 *// method (overloading) เขียนข้อมูลรายการสินค้าที่สั่งซื้อไปลงไฟล์ orders.txt โดยเอาข้อมูลจาก parameter orders มาเขียน  
 static void* write(vector<Order> orders, *int* totalNumbers, *float* totalAmount, string path = *R*"(C:\Users\ACER USER5949486\Desktop\CPP-project\txts\orders.txt)", *bool* showMessage = *false*){  
 *// ตัวแปรสำหรับเขียนไฟล์ข้อมูล* ofstream writeFile;  
 *// เปิดไฟล์เพื่อเขียนข้อมูล* writeFile.open(path, ios::app);  
 *// เช็คว่าสามารถเเปิดไฟล์ได้หรือไม่  
 if*(writeFile.is\_open()){  
 *int* i = 1;  
 *// สร้าง object time* Time time = Time();  
 *// เขียนเวลาล่าสุดที่เขียนในไฟล์ orders.txt* writeFile << "DATE: " << time.getDays() << " " << time.getDay() << " " << time.getMonths() << " " << time.getYear() << endl;  
 writeFile << "TIME: " << program::addZeroNumber(time.getHours()) << ":" << program::addZeroNumber(time.getMinutes()) << ":" << program::addZeroNumber(time.getSeconds()) << endl;  
 writeFile << "LIST:" << endl;  
 *// loop ข้อมูลตัวแปร orders  
 for*(Order order : orders){  
 *// เขียนข้อมูลสินค้าที่สั่งซื้อ* writeFile << i << ".) " << "PRODUCT\_NAME = " << order.getName() << ",\t\tID = " << order.getId() << ",\t\tPRICE = " << order.getPrice() << ",\t\t QUANTITY = " << order.getQuantity() << ",\t\t SUM = " << order.getSum() << ",\t\t BRAND = " << order.getBrand() << ",\t\t CATEGORY = " << order.getCategory() << endl;  
 i++;  
 }  
 *// เขียนสรุป จำนวนที่สั่งซื้อ และ ยอดจำนวนเงิน* writeFile << "CONCLUSION:" << endl;  
 writeFile << "Total number of products = " << totalNumbers << endl;  
 writeFile << "Total amount = " << totalAmount << " dollar." << endl;  
 *// เขียนเส้นตัดบรรทัดจบ  
 for*(*int* j = 1; j <= 170; j++){  
 writeFile << "-";  
 j == 170 && writeFile << endl;  
 }  
 *if*(showMessage) program::showSuccessfulMessage("Write file completed.");  
 } *else* {  
 *if*(showMessage) program::showErrorMessage("Cannot open file orders.txt to write data!");  
 }  
 writeFile.close();  
 }  
  
 *// method อัปเดตข้อมูลล่าสุดของไฟล์ data.txt และ ตัวแปรที่เก็บข้อมูลสินค้า data  
 static void* update(string path = *R*"(C:\Users\ACER USER5949486\Desktop\CPP-project\txts\data.txt)"){  
 write(path);  
 read(path);  
 }  
};  
  
*// สร้าง class หมวดหมู่สินค้า หรือ ประเภทสินค้าต่อไปนี้ โดยให้ subclass(หมวดหมู่สินค้า) สืบทอดคุณสมบัติทุกๆอย่าง ของ superclass(สินค้า)  
// Subclass  
class* Phone: *public* Product {  
*public*:  
 Phone(): Product(productCategories[0]){}  
};  
  
*// Subclass  
class* Tablet: *public* Product {  
*public*:  
 Tablet(): Product(productCategories[1]){}  
};  
  
*// Subclass  
class* Laptop: *public* Product {  
*public*:  
 Laptop(): Product(productCategories[2]){}  
};  
  
*// Subclass  
class* Computer: *public* Product {  
*public*:  
 Computer(): Product(productCategories[3]){}  
};  
  
*// Subclass  
class* Car: *public* Product {  
*public*:  
 Car(): Product(productCategories[4]){}  
};  
  
*// Subclass  
class* HealthAndBeauty: *public* Product {  
*public*:  
 HealthAndBeauty(): Product(productCategories[5]){}  
};  
  
*// Subclass  
class* Game: *public* Product {  
*public*:  
 Game(): Product(productCategories[6]){}  
};  
  
*// Subclass  
class* Bag: *public* Product {  
*public*:  
 Bag(): Product(productCategories[7]){}  
};  
  
*// Subclass  
class* ElectricalAppliance: *public* Product {  
*public*:  
 ElectricalAppliance(): Product(productCategories[8]){}  
};  
  
*// Subclass  
class* Pet: *public* Product {  
*public*:  
 Pet(): Product(productCategories[9]){}  
};  
  
*// Subclass  
class* Camera: *public* Product {  
*public*:  
 Camera(): Product(productCategories[10]){}  
};  
  
*// Subclass  
class* Shoes: *public* Product {  
*public*:  
 Shoes(): Product(productCategories[11]){}  
};  
  
*// Subclass  
class* Watch: *public* Product {  
*public*:  
 Watch(): Product(productCategories[12]){}  
};  
  
*// Subclass  
class* Sport: *public* Product {  
*public*:  
 Sport(): Product(productCategories[13]){}  
};  
  
*// Subclass  
class* MusicalInstrument: *public* Product {  
*public*:  
 MusicalInstrument(): Product(productCategories[14]){}  
};  
  
*// Subclass  
class* Furniture: *public* Product {  
*public*:  
 Furniture(): Product(productCategories[15]){}  
};  
  
*// Subclass  
class* Food: *public* Product {  
*public*:  
 Food(): Product(productCategories[16]){}  
};  
  
*// class Table สำหรับแสดงตารางสินค้าจากข้อมูลของตัวแปร data หรือ จากค่า argument ที่ส่งมา  
class* Table{  
*private*:  
 *// ตารางสำหรับแสดงสินค้า* char\_table table;  
 *int* number; *// ตัวเลขอันดับสินค้าใน column No  
public*:  
 *// constructor method สำหรับตั้งค่าเริ่มต้นในการสร้างตาราง* Table(){  
 *// ค่าเริ่มต้นของอันดับตัวเลขสินค้ามีค่าเป็น 1* number = 1;  
 *// เปลี่ยนเส้นขอบของตาราง* table.set\_border\_style(FT\_BASIC2\_STYLE);  
 *// จัดกึ่งกลางเนื้อหาของตาราง* table.set\_cell\_text\_align(text\_align::center);  
 *// จัดตำแหน่งของ column ชื่อสินค้าให้ชิดซ้าย* table.column(1).set\_cell\_text\_align(text\_align::left);  
 }  
  
 *// method แสดงตารางสินค้า  
 void* showTable(){  
 *if*(::data.size() == 0){  
 cout << on\_magenta << grey << "Out of stock!" << reset << endl;  
 } *else* {  
 cout << endl << "\t\t\t" << on\_bright\_white << grey << " List of all products " << reset << endl;  
 *// สร้างส่วนหัวของตารางโดยมีแต่ละ columds ตามนี้* table << header << "No" <<"Product" << "ID" << "$Price" << "Stock" << "Brand" << "Category" << endr;  
 *// loop เอาข้อมูลที่ได้มาแสดงผลทีละ row  
 for*(Product item : ::data){  
 table << number << item.getName() << item.getId() << item.getPrice() << item.getStock() << item.getBrand() << item.getCategory() << endr;  
 number++;  
 }  
 *// แสดงตาราง* cout << endl << table.to\_string() << endl;  
 }  
 }  
  
 *// method (overloading) สำหรับเแสดงตารางสินค้า orders สินค้าที่สั่งไป  
 void* showTable(vector<Order> orders){  
 *if*(orders.size() == 0){  
 cout << on\_magenta << grey << "Out of stock!" << reset << endl;  
 } *else* {  
 *// สร้างส่วนหัวของตารางโดยมีแต่ละ columds ตามนี้* table << header << "No" << "Product" << "ID" << "$Price" << "Quantity" << "Sum" << "Brand" << "Category" << endr;  
 *// loop เอาข้อมูลที่ได้มาแสดงผลทีละ row  
 for*(Order order : orders){  
 table << number << order.getName() << order.getId() << order.getPrice() << order.getQuantity() << order.getSum() << order.getBrand() << order.getCategory() << endr;  
 number++;  
 }  
 *// แสดงตาราง* cout << endl << table.to\_string() << endl;  
 }  
 }  
  
 *// method (overloading) สำหรับแสดงตารางสินค้าด้วยชื่อ Brand หรือ Category (ใช้คู่กับ method showProductCategory และ showProductBrand)  
 void* showTable(vector<Product> list){  
 *if*(list.size() == 0){  
 cout << on\_magenta << grey << "Out of stock!" << reset << endl;  
 } *else* {  
 cout << endl << "\t\t\t" << on\_bright\_white << grey << " List of all products " << reset << endl;  
 *// สร้างหัว columns* table << header << "No" << "Product" << "ID" << "$Price" << "Stock" << "Brand" << "Category" << endr;  
 *// loop ข้อมูลจาก parameter list โดยสร้างแต่ละ row  
 for*(Product item : list){  
 table << number << item.getName() << item.getId() << item.getPrice() << item.getStock() << item.getBrand() << item.getCategory() << endr;  
 number++;  
 }  
 *// แสดงตาราง* cout << endl << table.to\_string() << endl;  
 }  
 }  
};  
  
*// class ProductManagement มีหน้าที่จัดการเกี่ยวกับข้อมูลสินค้าภายในโปรแกรม  
class* ProductManagement {  
*public*:  
  
 *// method ในการตรวจสอบสินค้าว่ามีอยู่ในข้อมูลไหม ถ้ามีคืนค่า true ถ้าไม่ คืนค่า false  
 static bool* findProduct(string key){  
 *for*(Product item : ::data){  
 *// key เป็นได้ทั้ง รหัสสินค้า หรือ ชื่อสินค้าก็ได้  
 if*(to\_string(item.getId()) == key || item.getName() == key){  
 *return true*;  
 }  
 }  
 *return false*;  
 }  
  
 *// method (overloading) เฉพาะสำหรับตรวจสอบเลข id  
 static bool* findProduct(*int* id){  
 *for*(Product item : ::data){  
 *if*(item.getId() == id){  
 *return true*;  
 }  
 }  
 *return false*;  
 }  
  
 *// method ตรวจสอบว่าเป็นหมวดหมู่สินค้าที่ได้กำหนดไว้ไหม ถ้าใช้คืน true ถ้าไม่คืน false  
 static bool* isCategory(string category){  
 *for*(string c : productCategories){  
 *if*(c == category){  
 *return true*;  
 }  
 }  
 *return false*;  
 }  
  
 *// method ในการเช็คว่าข้อมูลตอนนี้ว่างเปล่าหรือไม่ (ไม่มีสินค้าอยู่ในไฟล์ data.txt) ถ้าว่างเปล่าคืน true ถ่าไม่ว่างเปล่าคืน false  
 static bool* isEmpty(){  
 *return* ::data.size() == 0;  
 }  
  
 *// method ในการตรวจสอบว่าค่า parameter ที่ส่งเข้ามานั้นเป็นเลขจำนวนเต็มบวกหรือไม่ ถ้าใช่คืน true ถ้าไม่คืน false  
 static bool* isPositiveNumber(*int* n){  
 *return* n > 0;  
 }  
  
 *// method แสดงรายการสินค้า  
 static void* showListProducts(){  
 *if*(isEmpty()){  
 cout << on\_magenta << grey << "Out of stock!" << reset << endl;  
 } *else* {  
 *int* number = 1;  
 *// แสดงตารางสินค้า* Table table = Table();  
 table.showTable();  
 }  
 }  
  
 *// method แสดงสินค้าเฉพาะสินค้าหมวดหมู่นั้น  
 static void* showProductCategory(){  
 string category;  
  
 cout << yellow << "Enter category:" << reset;  
 cin >> category;  
  
 *// เช็คว่ามีสินค้าหรือไม่  
 if*(isEmpty()){  
 cout << magenta << "Out of stock!" << reset << endl;  
 } *else* {  
 vector<Product> list;  
 *// เช็คว่าอยู่ในหมวดหมู่สินค้านั้นหรือไม่  
 if*(isCategory(category)){  
 *int* number = 1;  
 *// ตรวจสอบว่ามีหมวดหมู่สินค้านั้นอยู่ในคลัง  
 bool* inStock = *false*;  
 *// loop ข้อมูลสินค้า  
 for*(Product item : ::data){  
 *// แสดงสินค้าเฉพาะหมวดหมู่สินค้าที่เลือก  
 if*(item.getCategory() == category){  
 inStock = *true*;  
 list.push\_back(item);  
 }  
 }  
 *// ไม่มีสินค้าหมวดนี้อยู่ในคลังสินค้า  
 if*(!inStock){  
 string err[] = { " No product category ", "\"", category, "\"", " in stock." };  
 program::showErrorMessage(5, err);  
 *return*;  
 } *else* {  
 *// แสดงตารางสินค้าโดยส่ง argument list เข้าไป* Table table = Table();  
 table.showTable(list);  
 *// ล้างข้อมูลใน list ใหม่* list.clear();  
 }  
 *// ไม่อยู่ในหมวดหมู่สินค้าแสดงข้อความ error* } *else* {  
 string err[] = { "\"", category, "\"", " is not in categories of products." };  
 program::showErrorMessage(4, err);  
 }  
 }  
 }  
  
 *// method แสดงสินค้าเฉพาะสินค้าหมวดหมู่นั้น  
 static void* showProductBrand(){  
 string brand;  
  
 cout << yellow << "Enter brand name:" << reset;  
 cin >> brand;  
  
 *// เช็คว่ามีสินค้าหรือไม่  
 if*(isEmpty()){  
 cout << on\_magenta << grey << "Out of stock!" << reset << endl;  
 } *else* {  
 *int* number = 1;  
 vector<Product> list;  
 *// ตรวจสอบว่าหาแบรนด์สินค้าเจอ  
 bool* inStock = *false*;  
 *// loop ข้อมูลสินค้า  
 for*(Product item : ::data) {  
 *// แสดงสินค้าเฉพาะหมวดหมู่สินค้าที่เลือก  
 if* (item.getBrand() == brand && brand != "-") {  
 inStock = *true*;  
 list.push\_back(item);  
 }  
 }  
 *// ถ้าไม่พบแบรนด์สินค้านี้ ... ในคลังให้แสดงข้อความ error  
 if*(!inStock){  
 string err[] = { "This product brand ", "\"", brand, "\"", " was not found in stock!" };  
 program::showErrorMessage(5, err);  
 *return*;  
 } *else* {  
 *// แสดงตารางสินค้าโดยส่ง argument list เข้าไป* Table table = Table();  
 table.showTable(list);  
 *// ล้างข้อมูลใน list ใหม่* list.clear();  
 }  
 }  
 }  
  
 *// method เพิ่มสินค้า  
 static void* addProduct(*int* id){  
 string selectCategory;  
  
 *// แสดงหมวดหมู่สินค้าให้เลือกเพิ่มสินค้า* program::showProductCategories();  
  
 cout << endl << yellow << "Select product category:" << reset;  
 cin >> selectCategory;  
  
 *// ถ้ามีหมวดหมู่สินค้านั้นอยู่ในรายการ  
 if*(isCategory(selectCategory)){  
 ReceiveProduct p; *// สร้างตัวแปร p เพิ่อมารอรับข้อมูลสินค้าใหม่* Product newProduct; *// ตัวแปรที่เก็บ object ของสินค้าที่เพิ่มเข้ามาใหม่  
  
 // รับข้อมูลสินค้าใหม่ที่จะเพิ่ม* cout << yellow << "Product Name:" << reset;  
 cin >> p.name;  
 *// ตรวจสอบว่า name ว่าซ้ำกันไหม  
 if*(findProduct(p.name)){  
 program::showErrorMessage("The new product name must not be duplicated with the product that already has this name!");  
 *return*;  
 }  
 *// ห้ามตั้งชื่อสินค้าอักษรตัวแรกขึ้นต้นด้วยตัวเลข  
 else if*(isdigit(p.name.at(0))){  
 program::showErrorMessage("Do not name the product beginning with a number!");  
 *return*;  
 }  
  
 *// รับค่า parameter มาโดย id จะมีค่าเริ่มต้นที่สุ่มเอาไว้ให้* p.id = id;  
  
 cout << yellow << "Pricing:" << reset;  
 cin >> p.price;  
 *// ราคาต้องเป็นเลขจำนวนเต็มบวกเท่านั้น  
 if*(!isPositiveNumber(p.price)){  
 program::showErrorMessage("Invalid price, Please enter only positive number.");  
 *return*;  
 }  
  
 cout << yellow <<"If there is no product brand name, Enter -" << reset << endl;  
 cout << yellow << "Product brand name:" << endl;  
 cin >> p.brand;  
  
 *// สร้าง array ชื่อ products ทำหน้าที่เก็บหมวดหมู่สินค้าทั้งหมด* Product products[NUMBER\_CATEGORIES] = {  
 Phone(), Tablet(), Laptop(), Computer(), Car(), HealthAndBeauty(),  
 Game(), Bag(), ElectricalAppliance(), Pet(),  
 Camera(), Shoes(), Watch(), Sport(), MusicalInstrument(),  
 Furniture(), Food()  
 };  
 *// loop ข้อมูลใน array เพื่อเช็คว่าอยู่หมวดหมู่สินค้าไหน  
 for*(*int* i = 0; i < NUMBER\_CATEGORIES; i++){  
 *// เช็คหมวดหมู่สินค้าว่าตรงกันไหม  
 if*(selectCategory == products[i].getCategory()){  
 *// แก้ไขค่า สมาชิกใน array (แก้ไข ชื่อสินค้า รหัสสินค้า ราคา และ ชื่อแบรนด์)* products[i].setId(p.id);  
 products[i].setName(p.name);  
 products[i].setPrice(p.price);  
 *// ดึง substring ออกมาแล้วเช็คว่าเป็น - หรือไม่ ถ้าไม่มี brand พิมพ์ - แบรนด์จะมีความหมายคือไม่มีชื่อแบรนด์สินค้า แต่ถ้าไม่ได้พิมพ์ - แบรนด์จะเป็นค่าที่ผู้ใช้งานป้อนมา  
 if*(p.brand.at(0) != '-'){  
 products[i].setBrand(p.brand);  
 }  
 *// นำ newProduct เป็นค่า element ตัวนั้น* newProduct = products[i];  
 *// หยุด loop  
 break*;  
 }  
 }  
 *// นำ newProduct ที่ได้เพิ่มเข้าในรายการสินค้า data* ::data.push\_back(newProduct);  
 *// update รายการสินค้าล่าสุดของไฟล์ data.txt และ ข้อมูล data* File::update();  
 program::showSuccessfulMessage("Added a new product.");  
 }  
 *// ไม่อยู่ในหมวดหมู่สินค้าแสดงข้อความ error  
 else* {  
 string err[] = { "\"", selectCategory, "\"", " is not in categories of products" };  
 program::showErrorMessage(4, err);  
 }  
 }  
  
 *// method สำหรับเพิ่มจำนวนสินค้าใน stock  
 static void* addStockProduct(){  
 string input;  
 cout << yellow << "Enter the product name or product id:" << reset;  
 cin >> input;  
  
 *// ถ้ามีสินค้านั้นอย่ในข้อมูล  
 if*(findProduct(input)){  
 *int* number;  
 *int* index = 0;  
  
 cout << yellow <<"Amount:" << reset;  
 cin >> number;  
  
 *// จำนวนสินค้าที่เพิ่มเข้ามาต้องเป็นเลขจำนวนเต็มบวกเท่านั้น  
 if*(!isPositiveNumber(number)){  
 program::showErrorMessage("Invalid number, Please enter a positive number!");  
 *return*;  
 }  
  
 *// loop เช็คข้อมูลสินค้าทีละอัน เมื่อเช็คสินค้าเจอให้เพิ่มจำนวนสินค้าที่ระบุ  
 for*(Product item : ::data){  
 *// เช็ค ชื่อ หรือ id สินค้า ว่าตรงกันไหม  
 if*(input == to\_string(item.getId()) || input == item.getName()){  
 *// นำจำนวนค้าที่เหลือไปบวกกับจำนวนค้าที่เพิ่มเข้ามา  
 int* remain = number + item.getStock();  
 *// แก้ไขจำนวนใน stock* ::data.at(index).setStock(remain);  
 *// เมื่อเจอสินค้าที่ระบุแล้วให้หยุด loop  
 break*;  
 }  
 index++;  
 }  
 *// อัปเดตข้อมูล* File::update();  
 program::showSuccessfulMessage("Added new product quantity to stock");  
 }  
 *// ไม่มีสินค้านั้นอยู่ในข้อมูลแสดงช้อความ error  
 else* {  
 string err[] = { "\"", input, "\"", " is not in data!" };  
 program::showErrorMessage(4, err);  
 }  
 }  
  
 *// method ลบสินค้า  
 static void* deleteProduct() {  
 string input;  
  
 cout << yellow << "Enter the product name or product id:" << reset;  
 cin >> input;  
  
 *// ถ้ามีสินค้านั้นอยู่ในข้อมูล  
 if* (findProduct(input)) {  
 *int* index = 0;  
 *// loop ข้อมูลใน data  
 for* (Product item : ::data) {  
 *// เช็คชื่อ และ id สินค้าว่าตรงกันไหม  
 if* (to\_string(item.getId()) == input || item.getName() == input) {  
 *// ลบสินค้า(สมาชิกใน data)ออกจากตัวแปร data โดยเอาเลข index เป็นตัวบ่งบอกตำแหน่งของสมาชิกใน data* ::data.erase(::data.begin() + index);  
 *// เมื่อเจอสินค้าที่ระบุแล้วให้หยุด loop  
 break*;  
 }  
 index++;  
 }  
 program::showSuccessfulMessage("Successfully deleted product");  
 *// อัปเดตข้อมูล* File::update();  
 }  
 *// ไม่มีสินค้านั้นอยู่ในข้อมูลแสดงช้อความ error  
 else* {  
 string err[] = { "\"", input, "\"", " is not in data!" };  
 program::showErrorMessage(4, err);  
 }  
 }  
  
 *// method แก้ไขข้อมูลสินค้า  
 static void* editProduct() {  
 string input;  
  
 cout << yellow << "Enter the product name or product id:" << reset;  
 cin >> input;  
  
 *// ถ้ามีสินค้านั้นอยู่ในข้อมูล  
 if*(findProduct(input)){  
 *int* index = 0;  
 *// สร้างตัวแปรมารอรับข้อมูลที่ผู้ใช้งานป้อนเข้ามา* ReceiveProduct p;  
 *// คำตอบที่ผู้ใช้งานตอบมีแค่ y หรือ n เท่านั้น  
 typedef struct* {  
 string yn1, yn2, yn3, yn4, yn5;  
 } yesOrNo;  
 yesOrNo yn;  
  
 *// loop ข้อมูลสินค้าทั้งหมด  
 for* (Product item : ::data) {  
 *// เช็คว่าเป็นสินค้าชิ้นนั้น  
 if* (to\_string(item.getId()) == input || item.getName() == input) {  
 *// แสดงข้อมูลสินค้าเพื่อที่จะแก้ไข* cout << yellow << "You selected this product to edit." << reset << endl;  
 cout << on\_yellow << grey << " Product: " << item.getName() << "\tID: " << item.getId() << "\tPrice: " << item.getPrice() << "\tBrand: " << item.getBrand() << "\tCategory: " << item.getCategory() << " " << reset << endl << endl;  
  
 */\*  
 \* ถามว่าต้องการแก้ไขข้อมูลสินค้าในส่วนไหนบ้างโดยตอบ y และ n  
 \* ถ้าตอบ y ให้ดำเนินการแก้ไขข้อมูลในส่วนนั้น ถ้าตอบ n หรืออื่นๆคือผ่าน  
 \* ถามว่าต้องการแก้ไขชื่อสินค้าไหม  
 \* จะต้องตอบเป๋น y หรือ n เท่านั้นไม่งั้นจะ loop เรื่อยๆ  
 \*/  
 while*(*true*){  
 cout << cyan << "Do you want to edit the product name (y/n):" << reset;  
 cin >> yn.yn1;  
 *// เมื่อตอบ y  
 if*(program::yesOrNo(yn.yn1)){  
 cout << yellow << "New product name:" << reset;  
 cin >> p.name;  
 *// ตรวจสอบว่า name ที่แก้ไขว่าซ้ำกันกับข้อมูลที่มีแล้วไหม  
 if*(findProduct(p.name)){  
 string err[5] = { " Cannot edit to name ", "\"", p.name, "\"", " because the name is the same as an existing product name." };  
 program::showErrorMessage(5, err);  
 *// clear ข้อความเก่าเพื่อรับข้อความใหม่* yn.yn1.clear();  
 }  
 *// ห้ามตั้งชื่อสินค้าอักษรตัวแรกขึ้นต้นด้วยตัวเลข  
 else if*(isdigit(p.name.at(0))){  
 program::showErrorMessage("Do not name the product beginning with a number!");  
 *// clear ข้อความเก่าเพื่อรับข้อความใหม่* yn.yn1.clear();  
 } *else* {  
 *// แก้ไขชื่อสินค้า* ::data.at(index).setName(p.name);  
 *// แสดงข้อความว่าแก้ไขสำเร็จ* program::showSuccessfulMessage("edited product name.");  
 *// ออกจาก while loop  
 break*;  
 }  
 }  
 *// เมื่อตอบ n  
 else if*(!program::yesOrNo(yn.yn1) && tolower(yn.yn1.at(0)) == 'n' && yn.yn1.length() == 1){  
 *break*;  
 }  
 *// ไม่มีอยู่ในตัวเลือกทำให้วน loop ใหม่  
 else* {  
 program::showErrorMessage("Please answer only \"y\" or \"n\" !");  
 }  
 }  
  
 *while*(*true*){  
 *// ถามว่าต้องการแก้ไขชื่อรหัสสินค้าไหม* cout << cyan << "Do you want to edit the product id (y/n):" << reset;  
 cin >> yn.yn2;  
  
 *if*(program::yesOrNo(yn.yn2)){  
 cout << yellow << "New product id:" << reset;  
 cin >> p.id;  
 *// ตรวจสอบว่า name ที่แก้ไขว่าซ้ำกันกับข้อมูลที่มีแล้วไหม  
 if*(findProduct(p.id)){  
 string err[] = { " Cannot edit to id ", "\"", to\_string(p.id), "\"", " because the id is the same as an existing product id." };  
 program::showErrorMessage(5, err);  
 yn.yn2.clear();  
 cout << endl;  
 } *else* {  
 *// แก้ไขรหัสสินค้า* ::data.at(index).setId(p.id);  
 program::showSuccessfulMessage("edited product id.");  
 *break*;  
 }  
 } *else if*(!program::yesOrNo(yn.yn2) && tolower(yn.yn2.at(0)) == 'n' && yn.yn2.length() == 1){  
 *break*;  
 } *else* {  
 program::showErrorMessage("Please answer only \"y\" or \"n\" !");  
 }  
 }  
  
 *while*(*true*){  
 *// ถามว่าต้องการแก้ไขราคาสินค้าไหม* cout << cyan << "Do you want to edit the product price (y/n):" << reset;  
 cin >> yn.yn3;  
  
 *if*(program::yesOrNo(yn.yn3)){  
 cout << yellow << "New product price:" << reset;  
 cin >> p.price;  
 *// ตรวจสอบว่าเป็นเลขจำนวนเต็มบวกหรือไม่  
 if*(!isPositiveNumber(p.price)){  
 program::showErrorMessage("Invalid price, Please enter a positive number!");  
 *return*;  
 } *else* {  
 *// แก้ไขราคาสินค้า* ::data.at(index).setPrice(p.price);  
 program::showSuccessfulMessage("edited product price.");  
 *break*;  
 }  
 } *else if*(!program::yesOrNo(yn.yn3) && tolower(yn.yn3.at(0)) == 'n' && yn.yn3.length() == 1 ){  
 *break*;  
 } *else* {  
 program::showErrorMessage("Please answer only \"y\" or \"n\" !");  
 }  
 }  
  
 *while*(*true*){  
 *// ถามว่าต้องการแก้ไขหมวดหมู่สินค้าไหม* cout << cyan << "Do you want to edit the product category (y/n):" << reset;  
 cin >> yn.yn4;  
  
 *if*(program::yesOrNo(yn.yn4)){  
 *// แสดงหมวดหมู่สินค้าที่ให้เลือกแก้ไขหมวดหมู่สินค้า* program::showProductCategories();  
 cout << endl << yellow << "Select a product category to edit." << reset;  
 cout << yellow << "New product category:" << reset;  
 cin >> p.category;  
 *// ตรวจสอบว่าอยู่ในหมวดหมู่สินค้าที่ได้กำหนดไว้หรือไม่  
 if*(!isCategory(p.category)){  
 string err[4] = { "\"", p.category, "\"", " is not in categories of products!" };  
 program::showErrorMessage(4, err);  
 *return*;  
 } *else* {  
 *// แก้ไขหมวดหมู่สินค้า* ::data.at(index).setCategory(p.category);  
 program::showSuccessfulMessage("edited product category.");  
 *break*;  
 }  
 } *else if*(!program::yesOrNo(yn.yn4) && tolower(yn.yn4.at(0)) == 'n' && yn.yn4.length() == 1){  
 *break*;  
 } *else* {  
 program::showErrorMessage("Please answer only \"y\" or \"n\" !");  
 }  
 }  
  
 *while*(*true*){  
 *// ถามว่าต้องการแก้ไขชื่อแบรนด์สินค้าไหม* cout << cyan << "Do you want to edit the brand product (y/n):" << reset;  
 cin >> yn.yn5;  
  
 *if*(program::yesOrNo(yn.yn5)){  
 cout << yellow << "New brand product:" << reset;  
 cin >> p.brand;  
 *// แก้ไขแบรนด์สินค้า* ::data.at(index).setBrand(p.brand);  
 program::showSuccessfulMessage("edited product brand name.");  
 *break*;  
 } *else if*(!program::yesOrNo(yn.yn5) && tolower(yn.yn5.at(0)) == 'n' && yn.yn5.length() == 1){  
 *break*;  
 } *else* {  
 program::showErrorMessage("Please answer only \"y\" or \"n\" !");  
 }  
 }  
  
 *// เมื่อเจอสินค้าที่ระบุแล้วให้หยุด loop (ออกจาก for loop)  
 break*;  
 }  
 index++;  
 }  
 *// อัปเดตข้อมูล* File::update();  
 }  
 *// ไม่มีสินค้านั้นอยู่ในข้อมูลแสดงช้อความ error  
 else* {  
 string err[] = { "\"", input, "\"", " is not in data!" };  
 program::showErrorMessage(4, err);  
 }  
 }  
  
 *// method ในการขายสินค้า  
 static void* sellProducts(){  
 string input;  
 *bool* isRunning = *true*; *// ตัวแปรควบคุมการทำงาน while loop ถ้ามีค่า true แปลยังสามารถสั่งสินค้าต่อได้เรื่อยๆ ถ้า false หยุดดำเนินการสั่งซื้อ* vector<Order> orders; *// รายการ orders สินค้าที่สั่งซื้อทั้งหมด* cout << "Enter " << cyan << "\"end\"" << reset << " to exit the sale." << endl;  
  
 *// loop ไปเรื่อยๆจนกว่าผู้ใช้จะพิมพ์ตัว e  
 while*(isRunning){  
 cout << yellow << "Enter product name or product id:" << reset;  
 cin >> input;  
  
 *// เช็คว่าพิมพ์คำว่า end หรือไม่ ถ้าใช้ ให้ออกจากการขายสินค้า  
 if*(input == "end"){  
 *float* total = 0; *// จำนวนเงินทั้งหมด  
 int* quantity = 0; *// จำนวนสินค้าทั้งหมด  
 int* i = 0; *// ลำดับสินค้าที่สั่ง  
 // ออกจากการขายสินค้าและคำนวณราคาสินค้าทั้งหมด* isRunning = *false*;  
 *// ถ้ายังไม่มีการสั่งสินค้าไม่ต้องแสดงรายละเอียดการสั่งซื้อ  
 if*(orders.size() != 0){  
 *// คำนวณจำนวนเงินทั้งหมดที่สั่งสินค้า และ แสดงรายการสินค้าที่สั่งซื้อ  
 for*(Order item : orders){  
 *// คำนวณยอดเงินสินค้าต่อ 1 รายการ และแก้ไขยอดรวมสินค้านั้น  
 float* result = item.getQuantity() \* item.getPrice();  
 item.setSum(result);  
 *// แก้ไขค่ายอดรวมของสินค้าของแต่ละสินค้า* orders.at(i).setSum(item.getSum());  
 *// คำนวณเงินที่ต้องจ่ายทั้งหมดที่สั่งสินค้ามา* total += item.getSum();  
 *// เพิ่มจำนวนสินค้า* quantity += item.getQuantity();  
 i++;  
 }  
 cout << endl << "\t\t" << on\_bright\_white << grey << " The products you ordered. " << reset << endl;  
 *// สร้าง object table เพื่อจะแสดงตารางสินค้า* Table table = Table();  
 *// แสดงตารางรายละเอียดสินค้าและสรุปการสั่งซื้อสินค้า* table.showTable(orders);  
 *// แสดงจำนวนเงินทั้งหมดที่ต้องจ่าย* cout << endl << cyan << "Total number of products = " << quantity << reset << endl;  
 cout << cyan << "Total amount = " << total << " dollar." << reset << endl;  
 *// เขียนข้อมูลลงในไฟล์ orders.txt* File::write(orders, quantity, total);  
 *// ลบรายการสินค้าทั้งหมดที่สั่่ง* orders.clear();  
 *// อัปเดตข้อมูล* File::update();  
 }  
 }  
 *// ดำเนินการสั่งสินค้าต่อ  
 else* {  
 *// เช็คว่า ชื่อ หรือ id ที่พิมพ์มาอยู่ใน data หรือไม่  
 if*(findProduct(input)){  
 *int* j = 0; *// ตัวระบุเลข index ของ data  
 int* quantity; *// จำนวนสินค้า  
 // loop ข้อมูลในตัวแปร data  
 for*(Product item : ::data){  
 *// เช็ค ชื่อ หรือ id ว่าตรงกับสินค้าที่เลือก  
 if*(to\_string(item.getId()) == input || item.getName() == input){  
 *// เก็บ order สินค้าที่สั่ง (ส่งค่า args ให้ class Order เพื่อให้ค่า args ส่งไปยัง constructor ของ class Product ไว้จัดเก็บข้อมูลสินค้าที่สั่ง)* Order order = Order(item.getId(), item.getName(), item.getPrice(), item.getBrand(), item.getCategory()); *// order สินค้า 1 รายการ  
  
 // รับค้าจำนวนสินค้าที่สั่ง* cout << yellow << "Quantity:" << reset;  
 cin >> quantity;  
  
 order.setSum(0); *// ยอดรวมสินค้านั้นมีค่าเริ่มต้นเป็น 0* order.setQuantity(quantity); *// จำนวนสินค้าที่สั่ง  
  
 // จำนวนสินค้าต้องเป็นเลขจำนวนเต็มบวก  
 if*(!isPositiveNumber(order.quantity)) {  
 program::showErrorMessage("Invalid quantity, Please enter a positive number!");  
 isRunning = *false*;  
 *return*;  
 }  
 *// สินค้าในคลังหมดไม่สามารถสั่งได้  
 else if*(item.getStock() == 0){  
 cout << red << "This product " << "\"" << item.getName() << "\"" << " is out of stock." << reset << endl;  
 }  
 */\* เงื่อนไข  
 \* จำนวนที่สั่งต้องน้อยกวาหรือเท่ากับสินค้าในคลัง (จำนวนที่สั่งต้องไม่มากเกินจำนวนสินค้าในคลัง)  
 \* สินค้าในคลังต้องไม่หมด (ถ้าสินค้าในคลังหมดไม่สามารถสั่งได้)  
 \* ประมาณจำนวนสินค้านั้นในคลังก่อนเมื่อลองหักลบแล้วจำนวนสินค้าในคลังต้องไม่ติดลบ (ไม่สามารถสั่งเกินจำนวนสินค้าในคลังได้)  
 \*/  
 else if*((order.getQuantity() <= item.getStock()) && (item.getStock() != 0) && ((item.getStock() - order.getQuantity()) >= 0)){  
 *// จำนวนที่เหลือของสินค้าในคลัง โดยหักลบกับจำนวนสินค้าที่สั่ง  
 int* remain = item.getStock() - order.getQuantity();  
 *// เปลี่ยนค่าใน stock มีจำนวนสินค้าที่เหลือตาม remain* ::data.at(j).setStock(remain);  
  
 *// ต้องการตรวจสอบข้อมูลสินค้าที่สั่ง (order) นั้นเป็นสินค้าชิ้นเดียวกับที่เราพึ่งสั่งไป (สินค้าอันเดียวกับใน orders)  
 // ให้เก็บข้อมูลนั้นเป็นข้อมูลเดียวกันโดยไม่ต้องเพิ่ม element ตัวใหม่เข้าไป  
 int* k = 0; *// เลข index  
 bool* isDuplicate = *false*; *// เป็นสินค้าซ้ำกันหรือไม่  
 // loop ข้อมูลเพื่อเช็คว่าซ้ำกันไหม  
 for*(Order o: orders){  
 *// ถ้าชื่อหรือ id ซ้ากับสินค้าที่สั่งอยู่ให้ isDuplicate เป็น true  
 if*(o.getName() == order.getName() || o.getId() == order.getId()){  
 isDuplicate = *true*;  
 *break*;  
 }  
 k++;  
 }  
 *// ถ้าข้อมูลนั้นซ้ำกันให้เพิ่มจำนวนสินค้าที่สั่งอยู่  
 if*(isDuplicate){  
 *// จำนวนสินค้าที่เคยสั่งรวมกับจำนวนสินค้าที่พึ่งสั่ง  
 int* addQuantity = orders.at(k).getQuantity() + order.getQuantity();  
 orders.at(k).setQuantity(addQuantity);  
 }  
 *// ถ้าข้อมูลไม่ซ้ำกันให้ orders เพิ่ม element(สินค้า) ตัวใหม่เข้าไป  
 else* {  
 *// เพิ่มสินค้าเข้าใน orders ที่สั่ง* orders.push\_back(order);  
 }  
 *// เขียนไฟล์ข้อมูล* File::write();  
 }  
 *// สั่งสินค้าเกินจำนวนในคลัง  
 else* {  
 program::showErrorMessage(" The quantity of products ordered is greater than the quantity of products in stock!");  
 isRunning = *false*;  
 *return*;  
 }  
 }  
 j++;  
 }  
 }  
 *// ชื่อ หรือ id ที่ใส่มาไม่อยู่ในข้อมูลแสดงข้อความ error  
 else* {  
 string err[] = { "\"", input, "\"", " is not in data!" };  
 program::showErrorMessage(4, err);  
 }  
 }  
 }  
 }  
};  
  
*// สร้างตัวแปรเป็น enum เพื่อเก็บหมายเลขการทำงานของโปรแกรมไว้  
enum* options {  
 ShowTable = 1,  
 ShowCategory,  
 ShowBrand,  
 Sell,  
 Add,  
 Delete,  
 Edit,  
 AddStock,  
 Clear,  
 Exit  
};  
  
*int* main(){  
 *// เริ่มโปรแกรมให้อ่านข้อมูลจากไฟล์ data.txt แล้วมาเก็บไว้ในตัวแปร data* File::read();  
 *// ตัวเลือกที่ผู้ใช้งานเลือกว่าจะใช้งานคำสั่งอะไร  
 int* select;  
 *// วน loop ไปเรื่อยๆเพื่อรอให้ผู้ใช้งานป้อนตัวเลขให้โปรแกรมทำงานตามหมายเลขนั้น  
 while*(*true*){  
 *// แสดงตัวเลือกที่ผู้ใช้จะเลือกใช้งานคำสั่งของโปรแกรม โดยผู้ใช้งานจะต้องป้อนตัวเลข 1 - 10 เข้ามา* program::showOptions();  
 cin >> select;  
  
 *// แสดงรายการสินค้าทั้งหมด  
 if*(select == ShowTable){  
 ProductManagement::showListProducts();  
 }  
 *// แสดงเฉพาะหมวดหมู่สินค้าที่เลือก  
 else if*(select == ShowCategory){  
 ProductManagement::showProductCategory();  
 }  
 *// แสดงเฉพาะแบรนด์สินค้าที่เลือก  
 else if*(select == ShowBrand){  
 ProductManagement::showProductBrand();  
 }  
 *// สั่งซื้อสินค้า  
 else if*(select == Sell){  
 ProductManagement::sellProducts();  
 }  
 *// เพิ่มสินค้า  
 else if*(select == Add){  
 *// เมื่อเพิ่มสินค้าใหม่เข้าไปจะ gen id ให้อัตโนมัติโดยที่ id ไม่ซ้ำกัน  
 int* id = program::generateId(1000,10000);  
 ProductManagement::addProduct(id);  
 }  
 *// ลบสินค้า  
 else if*(select == Delete){  
 ProductManagement::deleteProduct();  
 }  
 *// แก้ไขสินค้า  
 else if*(select == Edit){  
 ProductManagement::editProduct();  
 }  
 *// เพิ่มจำนวนสินค้าในคลัง  
 else if*(select == AddStock){  
 ProductManagement::addStockProduct();  
 }  
 *// ล้างหน้าจอ  
 else if*(select == Clear){  
 system("cls");  
 }  
 *// ออกจากโปรแกรม  
 else if*(select == Exit){  
 cout << endl << on\_bright\_white << grey << "Exit program." << reset << endl;  
 *break*;  
 }  
 *// ไม่มีในตัวเลือก  
 else* {  
 string err[] = { "\"", to\_string(select), "\"", " is not available, Please select a number between 1 - 10." };  
 program::showErrorMessage(4, err);  
 }  
 }  
  
 *return* 0;  
}  
  
*namespace* program {  
 *// function แสดงตัวเลือกการทำงานของโปรแกรม  
 void* showOptions(){  
 *// สร้าง array ไว้เก็บชุดความหมายของคำสั่ง* string meaningOfCommands[10] = { "Show list of all products", "Show list of product category", "Show list of product brand",  
 "Sell proudcts", "Add product", "Delete product", "Edit product", "Add product to stock",  
 "Clear console screen", "Exit program" };  
 *// สร้าง object time ไว้แสดงเวลาสุดทุกครั้งที่ใข้งาน* Time time = Time();  
  
 cout << endl << blue << "Product management program" << reset << endl;  
 cout << "Current Time " << cyan;  
 cout << addZeroNumber(time.getHours()) << ":" << addZeroNumber(time.getMinutes()) << ":" << addZeroNumber(time.getSeconds()) << reset << endl << endl;  
  
 string blank = " ";  
 *// loop แสดงข้อความตัวเลือกของโปรแกรมโดยสีพื้นหลังจะต้องเท่ากัน  
 for*(*int* i = 0; i < 10; i++){  
 *if*(i == 9) blank.clear();  
 cout << on\_bright\_white << grey << " " << i + 1 << ". " << blank << reset << on\_blue << grey << " " << meaningOfCommands[i];  
 *// เพิ่มข้อความเปล่าเพื่อช่องว่าให้สีพื้นหลังนั้นแสดงเท่ากัน  
 for*(*int* j = meaningOfCommands[i].length(); j <= 30; j++){  
 cout << " ";  
 }  
 cout << reset << endl;  
 }  
 cout << on\_yellow << grey << "Enter a number:" << reset;  
 }  
  
 *// function ในการแสดงข้อความ error  
 void* showErrorMessage(string message){  
 *// แสดงข้อความตามสีรูปแบบทีกำหนด* cout << on\_bright\_grey << red << " Error: " << reset << on\_red << grey << " " << message << " " << reset << endl;  
 }  
  
 *// function (overloading) แสดงข้อความ error ที่ต้องการแนบตัวแปรเข้าไปด้วย  
 void* showErrorMessage(*int* size, string message[]){  
 cout << on\_bright\_grey << red << " Error: " << reset << on\_red << grey << " " << on\_red << grey;  
 *for*(*int* i = 0; i < size; i++){  
 cout << message[i];  
 }  
 cout << " " << reset << endl;  
 }  
  
 *// function ในการแสดงข้อความที่ทำสำเร็จ  
 void* showSuccessfulMessage(string message){  
 *// แสดงข้อความตามสีรูปแบบทีกำหนด* cout << on\_bright\_grey << green << " Successfully: " << reset << on\_green << grey << " " << message << " " << reset << endl;  
 }  
  
 *// fucntion ในการแสดงหมวดหมู่สินค้าทั้งหมด  
 void* showProductCategories(){  
 cout << "Product categories ";  
 *char* comma = ',';  
 *int* count = 0;  
 *// แสดงหมวดหมู่ของสินค้าที่สามารถเพิ่มได้  
 for*(string category : productCategories){  
 cout << green << "\"" << category << "\"" << reset;  
 *if*(count != NUMBER\_CATEGORIES - 1) cout << comma << ' ';  
 count++;  
 }  
 }  
  
 *// function ในการสร้างเลข id โดยที่เลข id จะสุ่มเลขอยู่ระหว่าง from ถึง to  
 int* generateId(*int* from, *int* to){  
 *// ตั้งค่าการสุ่ม* srand(time(NULL));  
 *// ค่า id ที่ได้จากการสุ่ม มีเลข 4 หลัก  
 int* id;  
 *// สุ่มเลขระหว่าง from ถึง to จะได้ 4 หลัก* id = rand() % to + from;  
 *// ตรวจสอบเลข id นั้นซ้ำกับเลข id สินค้าอื่นหรือไม่  
 bool* isDuplicate = ProductManagement::findProduct(id);  
  
 *// ถ้าเลข id นั้นซ้ำกับเลข id สินค้าอื่น  
 if*(isDuplicate){  
 *// loop ไปเรื่อยๆจนกว่าเลข id จะไม่ซ้ำ  
 while*(isDuplicate){  
 *int* i = 1; *// นับเลขว่าถึง loop รอบสุดท้ายหรือยัง* id = rand() % to + from; *// สุ่มเลขใหม่  
 // ตรวจสอบเลข id  
 for*(Product item : ::data){  
 isDuplicate = ProductManagement::findProduct(id);  
 *// เมื่อถึง loop รอบสุดท้ายของข้อมูล  
 if*(i == ::data.size()){  
 *// ถ้าเลข id นั้นไม่ซ้ำให้ออกจาก while loop แต่ถ้าเลข id นั้นยังซ้ำอยู่ก็ loop ไปเรื่อยๆจนกว่าจะไม่ซ้ำ  
 if*(!isDuplicate) isDuplicate = *false*;  
 }  
 i++;  
 }  
 }  
 }  
  
 *return* id;  
 }  
  
 *// funciton ในการเพิ่มเลข 0 นำหน้าตัวเลขเมื่อเลขนั้นน้อยกว่า 10* string addZeroNumber(*int* num){  
 string newNum = "0";  
 newNum.append(to\_string(num));  
 *return* num < 10 ? newNum : to\_string(num);  
 }  
  
 *// function ในการรับข้อมูลนั้นเป็น y หรือ n ไหม ถ้าเป็น y คืน true ถ้าเป็น n หรือตัวอื่นๆจะคืน false  
 bool* yesOrNo(string yn){  
 *// ตัวอักษรนั้นต้องเป็น 1 ตัวอักษร  
 if*(yn.length() == 1){  
 *return* tolower(yn.at(0)) == 'y';  
 } *else* {  
 *return false*;  
 }  
 }  
}

**เกี่ยวกับโปรแกรม**

โปรเจคนี้ทำเกี่ยวกับ โปรแกรมจัดการสินค้าใน ร้านสินค้า , ร้านขายของ , ร้านสะดวกซื้อ หรือ ร้านอื่นๆ โปรแกรมนี้ได้นำหลักการของ OOP มาใช้ในการพัฒนาโปรแกรมตามที่อาจารย์ได้ใส่ในข้อกำหนดไว้เรียบร้อยเสร็จสมบรูณ์ โปรแกรมนี้จึงมีหลักการพื้นฐานของ OOP ทั้งสิ้น ในส่วนการใช้งานโปรแกรมนี้เมื่อรันโปรแกรมผ่าน CLI ของโปรแกรมจะมีตัวเลือกคือตัวเลขโดยแต่ละตัวเลขคือ ความหมายของคำสั่งที่ให้โปรแกรมทำงานนั้นๆเมื่อผู้ใช้งานป้อนตัวเลขเข้าไปในโปรแกรม โปรแกรมก็จะแสดงหรือโต้ตอบกับเรา โดยจะแสดงข้อมูลหรือทำงานบางอย่างตามที่ป้อนตัวเลขนั้นเข้าไป สามารถใช้งานโปรแกรมได้ผ่านตามลิ้งค์ repository ของโปรเจคนี้ในหน้า README.md จะแสดงขั้นตอนการใช้งานและวิธีติดตั้งไว้เรียบร้อยแล้ว

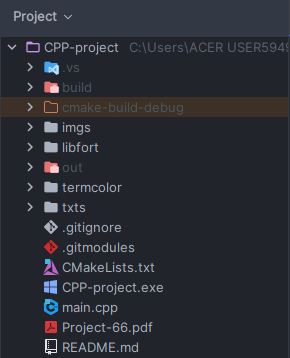


**ฟีเจอร์ของโปรแกรม**

1. ระบบเพิ่มสินค้า
2. ระบบลบสินค้า
3. ระบบแก้ไขสินค้า
4. ระบบสั่งซื้อสินค้า
5. ระบบเพิ่มสินค้าใน stock
6. ระบบแสดงตารางสินค้า (แสดงสินค้าทั้งหมด , แสดงหมวดหมู่สินค้า , แสดงแบรนด์สินค้า)
7. ระบบอ่านเขียนไฟล์ข้อมูล
8. ระบบเขียนไฟล์ข้อมูลสรุปการสั่งซื้อสินค้า

**โครงสร้างของโปรเจค**

โปรเจคนี้ประกอบด้วยไฟล์และโฟลเดอร์อยู่หลายอันด้วยกันโดยเราจะมาอธิบายเฉพาะส่วนที่สำคัญกัน



ไฟล์ main.cpp คือไฟล์โค้ดหลักของโปรแกรม

โปรแกรม CPP-project.exe คือโปรแกรมที่ได้จากการ compile ไฟล์ main.cpp เพื่อใช้งานโปรแกรม

โฟลเดอร์ txts จะเก็บไฟล์ข้อความไว้อยู่ 2 ไฟล์คือ

* 1. ไฟล์ data.txt ไว้เก็บข้อมูลสินค้าทั้งหมด
  2. ไฟล์ orders.txt ไว้เก็บรายการสินค้าที่เคยถูกสั่งซื้อไปโดยจัดเก็บข้อมูลการสั่งซื้อสินค้าทั้งหมด

ไฟล์ CMakelists.txt เป็นไฟล์ setting ของโปรเจคนี้

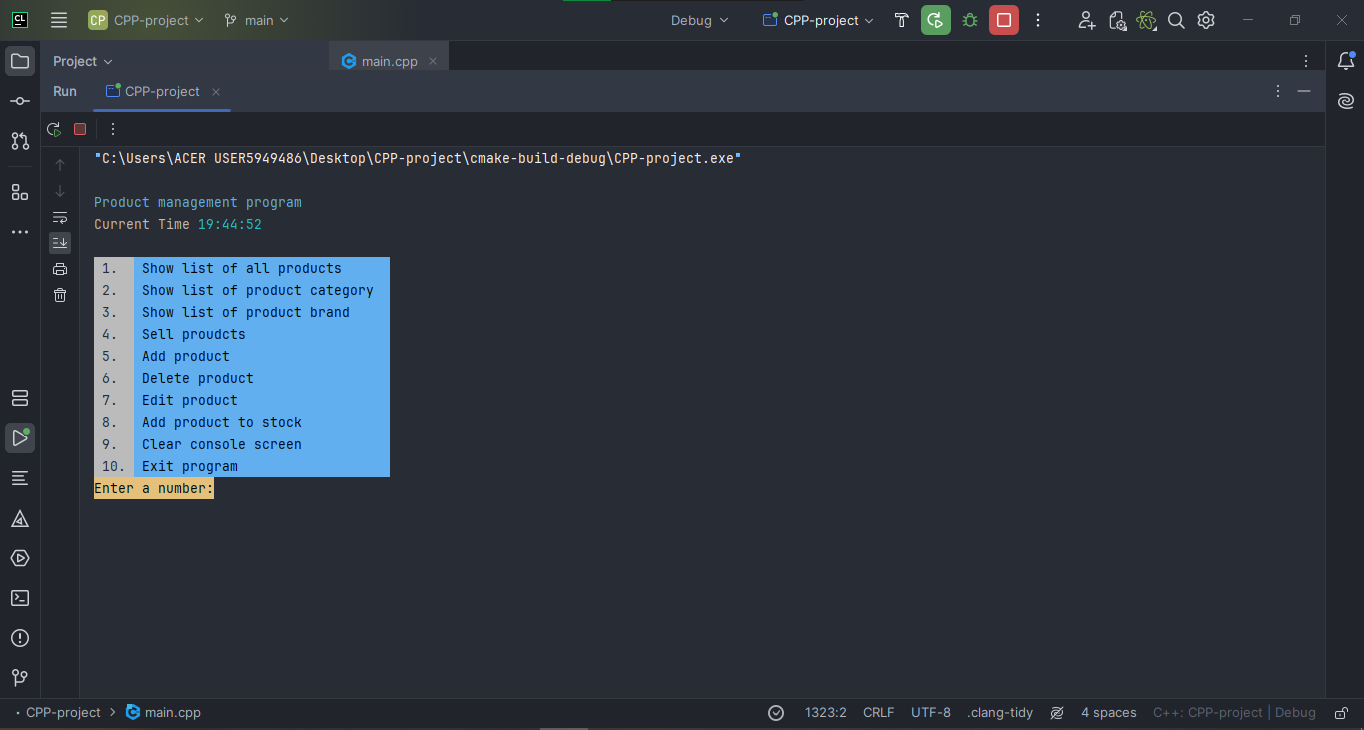
โฟลเดอร์ imgs ไว้เก็บ assets ต่างๆไว้ใช้เขียนไฟล์ README.md

ไฟล์ README.md ไว้อธิบายรายละเอียดเกี่ยวกับเจคนี้

โฟลเดอร์ termcolor และ libfort เป็น folder libraries ถูกไว้เรียกใช้งานในไฟล์ main.cpp

**อธิบายผลลัพธ์และสิ่งที่ต้องป้อนเข้าไปในโปรแกรม**

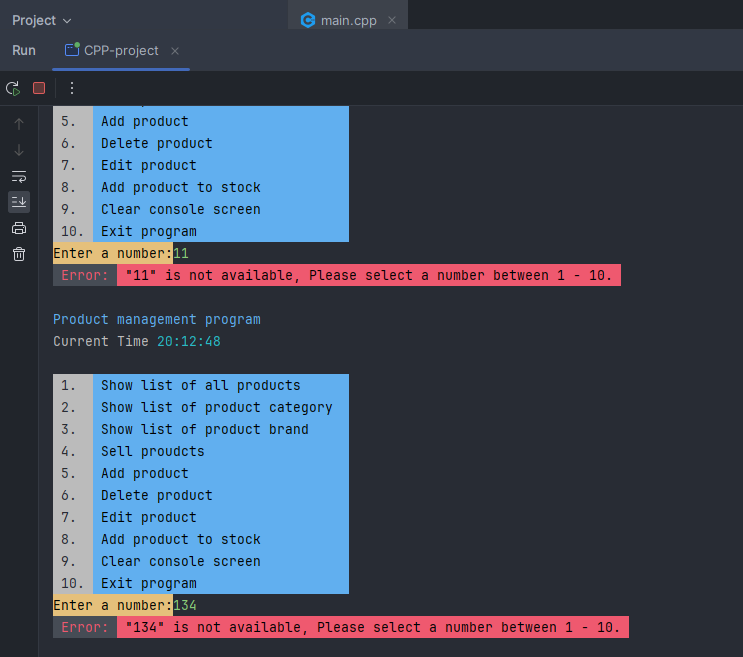
โดยเริ่มจากรันโปแกรม main.cpp จะได้โปรแกรมหน้าตาประมาณนี้ตอนเริ่มต้น



โดยเมื่อรันโปรแกรมโปรแกรมจะรอคำสั่งจากผู้ใช้งานให้ป้อนตัวเลขระหว่างเลข 1 ถึง 10 โดยแต่ละเลขจะเป็นความหมายของแต่ละคำสั่งโดยมีความหมายของคำสั่งเป็นดังนี้

1. แสดงรายการสินค้าทั้งหมด
2. แสดงรายการหมวดหมู่สินค้า
3. แสดงรายการแบรนด์สินค้า
4. สั่งซื้อสินค้า
5. เพิ่มสินค้า
6. ลบสินค้า
7. แก้ไขสินค้า
8. เพิ่มจำนวนสินค้าใน stock
9. ล้างหน้าจอโปรแกรม
10. ออกจากโปรแกรม

หากป้อนเลขอื่นที่นอกจากเลขที่ระบุไว้ในนี้โปรแกรมจะไม่เจอคำสั่งใดๆแล้วจะแสดงให้ป้อนตัวเลขใหม่



**ข้อควรระวัง** การที่จะพิมพ์อะไรก็ได้นั้นในโปรแกรมนี้มีข้อจำกัดอยู่ด้วยการเขียนโปรแกรมภาษา C++ เราไม่สามารถรับมือกับข้อมูลที่เป็นข้อความเมื่อเราต้องการให้ผู้ใช้งานป้อนเป็นตัวเลขมันจะเกิดข้อผิดพลาดขึ้นหากเรารอข้อมูลที่เป้นชนิดตัวเลขแต่ผู้ใช้งานดันป้อนข้อมูลเป็นชนิดข้อความจะเกิดข้อผิดพลาดขึ้นคือโปรแกรมจะ bug แสดงผลไปเรื่อยๆจนคอมพิวเตอร์เครื่องเราค้าง เพราะฉะนั้นควรอ่านข้อความให้เข้าใจก่อนว่าโปรแกรมต้องการให้พิมพ์ข้อมูลเป็นชนิดอะไร

โดยเราจะเริ่มต้นที่คำสั่งแรกกันเลยคือ แสดงรายการสินค้าทั้งหมดรายการที่แสดงออกมาจะมาจากไฟล์ข้อมูล data.txt จะแสดงออกมาเป็นตารางสินค้า

**คำสั่งที่ 1 แสดงรายการสินค้าทั้งหมด**

พิมพ์เลข 1 เข้าไปในโปรแกรมจะมีข้อความแสดงขึ้นดังภาพต่อไปนี้

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ภาพหน้าจอ, ข้อความ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

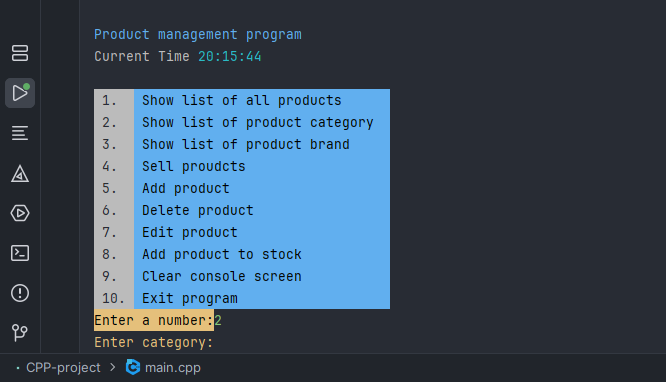
รูปภาพประกอบด้วย ภาพหน้าจอ, ข้อความ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

ตารางจะแสดงข้อมูลรายละเอียดดังภาพคือมี ชื่อสินค้า , เลขรหัสสินค้า , ราคาสินค้า , จำนวนสินค้าใน stock , แบรนด์สินค้า และ หมวดหมู่สินค้า โดยความยาวของตารางสินค้าจะขึ้นอยู่กับจำนวนข้อมูลในไฟล์ data.txt

**คำสั่งที่ 2 แสดงรายการหมวดหมู่สินค้า**

พิมพ์เลข 2 เข้าไปในโปรแกรมจะมีข้อความแสดงขึ้นดังภาพต่อไปนี้



ต้องพิมพ์ชื่อหมวดหมู่สินค้า

ในส่วนนี้จะต้องพิมพ์ข้อความหมวดหมู่สินค้าที่เราต้องการค้นหาสินค้านั้นโดยมีข้อความที่สามารถพิมพ์ได้ดังนี้

phone , tablet , laptop , computer , car , health and beauty , game , bag , electrical appliance, pet , camera , shoes , watch

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ภาพหน้าจอ, ข้อความ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

เมื่อป้อนข้อความดังกล่าวก็จะแสดงข้อมูลที่เป็นเฉพาะหมดวหมู่สินค้านั้นออกมา

รูปภาพประกอบด้วย ข้อความ, อิเล็กทรอนิกส์, ภาพหน้าจอ, ซอฟต์แวร์

คำอธิบายที่สร้างโดยอัตโนมัติ รูปภาพประกอบด้วย ข้อความ, อิเล็กทรอนิกส์, ภาพหน้าจอ, ซอฟต์แวร์

คำอธิบายที่สร้างโดยอัตโนมัติ

ถ้าหากป้อนข้อความที่ไม่ได้เป็นหมวดหมู่สินค้าจะแสดงข้อความ error ดังภาพ

ถ้าหมวดหมู่สินค้านั้นยังไม่มีอยู่ในไฟล์ data.txt จะแสดงข้อความว่าสินค้าหมวดหมู่นั้นยังไม่มีในข้อมูล

**คำสั่งที่ 3 แสดงรายการแบรนด์สินค้า**

พิมพ์เลข 3 เข้าไปในโปรแกรมจะมีข้อความแสดงขึ้นดังภาพต่อไปนี้

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

ต้องพิมพ์ชื่อแบรนด์สินค้า

เมื่อพิมพ์แบรนด์สินค้าจะค้นหาข้อมูลแบรด์สินค้านั้นและแสดงออกมา

รูปภาพประกอบด้วย ข้อความ, อิเล็กทรอนิกส์, ภาพหน้าจอ, ซอฟต์แวร์

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

ไม่มีแบรนด์สินค้านั้นที่เก็บข้อมูลอยู่จะแสดงข้อความ error ตามดังภาพ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, แสดง, ตัวอักษร

คำอธิบายที่สร้างโดยอัตโนมัติ

**คำสังที่ 4 สั่งซื้อสินค้า**

พิมพ์เลข 4 เข้าไปในโปรแกรมจะมีข้อความแสดงขึ้นดังภาพต่อไปนี้

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

บอกผู้ใช้งานว่าถ้าหากสั่งซื้อสินค้าพอแล้วจบแล้วให้พิมพ์ตัว

“end” เพื่อออกจากการสั่งซื้อสินค้า

พิมพ์ชื่อสินค้า หรือ รหัสสินค้า ก็ได้เพื่อค้นหาสินค้าที่ผู้ใช้งานต้องการจะซื้อ

ผู้ใช้งานจะไม่รู้เลยว่าสินค้าที่เราขายในโปรแกรมนี้นั้นมีสินค้าอะไรเลยบ้างเราสามารถใช้คำสั่งหมายเลข 1 และ หมายเลข 4 ควบคู่กันกับการสั่งซื้อสินค้าได้เพื่อให้ผู้ใช้งานรู้ว่ามีสินค้าอะไรบ้างที่ขาย

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

พิมพ์ end เพื่อออกจากการสั่งซื้อแล้วกดคำสั่งหมายเลข 1 ใหม่เพื่อแสดงรายการข้อมูลสินค้าก่อนแล้วกดคำสั่งหมายเลข 4 อีกรอบจเพื่อจะดูได้ว่ามีอะไรสามารถสั่งซื้อได้บ้าง

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

ตอนนี้เราสามารถดูสินค้าได้แล้วว่าจะพิมพ์อะไรได้แล้วบ้างทดลองสั่งซื้อสินค้าดู

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

จำนวนสินค้าที่สั่งซื้อไป

สามารถพิมพ์รหัสสินค้าหรือเลข id แทนชื่อสินค้าได้เลยโดยเลข id จะมีความยาวเพียง 4 หลักจึงง่ายต่อการพิมพ์กว่าชื่อสินค้ายาวๆสามารถใช้แทนกันได้

สามารถพิมพ์ชื่อสินค้าได้เลยแต่ชื่อต้อง

ตรงกับสินค้าในตาราง

รูปภาพประกอบด้วย ข้อความ, อิเล็กทรอนิกส์, ภาพหน้าจอ, ซอฟต์แวร์

คำอธิบายที่สร้างโดยอัตโนมัติ

ข้อผิดพลาดที่สามารถเกิดขึ้นจากการสั่งซื้อสินค้าถูกรับมือไว้หมดเรียบร้อยแล้ว

ไม่มีชื่อสินค้านั้นอยู่ตริงในข้อมูลแสดงข้อความ error

เลข id ที่ไม่มีอยู่จริงจะไม่มีเลข id นี้อยู่ในข้อมูลก้แสดงข้อความ error

จำนวนสินค้าไม่สามรถเป็นเลขจำนวนเต็มศูนย์ได้

รูปภาพประกอบด้วย ข้อความ, อิเล็กทรอนิกส์, ภาพหน้าจอ, ซอฟต์แวร์

คำอธิบายที่สร้างโดยอัตโนมัติ

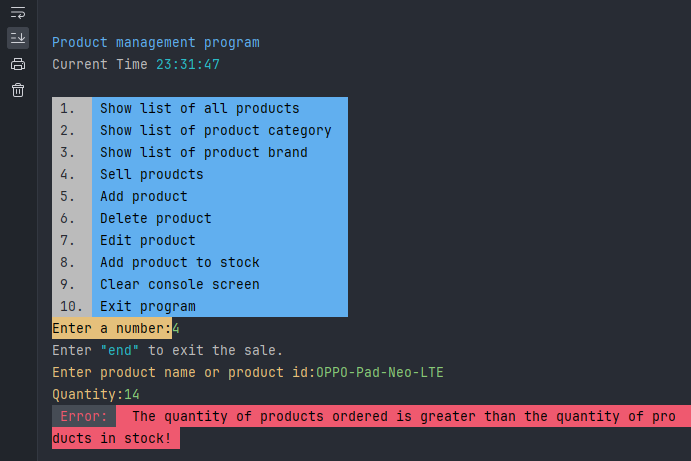
และเมื่อใส่เลขผิดพลาดจะออกจากการสั่งซื้อทันทีและเริ่มที่หน้าโปรแกรมใหม่

จำนวนสินค้าไม่สามรถเป็นเลขจำนวนเต็มลบได้ต้องเป็นจำนวนเต็มบวกเท่านั้น

สมมติ ถ้าลูกค้าต้องการซื้อโทรศัพท์รุ่นนี้แต่สั่งเกินจำนวนสินค้าใน stock สินค้าที่เรามีอยู่เรา(สั่งสินค้าเกินจำนวนที่มีอยู่ในคลังสินค้า) โปรแกรมจะไม่อณุญาติให้ทำการสั่งซื้อโทรศัพท์นี้ได้ เนื่องจากไม่สามารถสั่งซื้อสินค้าเกินจำนวนสินค้าใน stock ได้

รูปภาพประกอบด้วย ภาพหน้าจอ, ข้อความ, ไลน์, ตัวอักษร

คำอธิบายที่สร้างโดยอัตโนมัติ



สั่งสินค้าเกินจำนวนสินค้าในคลังไม่สามารถสั่งสินค้านี้ได้และแสดงข้อความerror

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, แสดง

คำอธิบายที่สร้างโดยอัตโนมัติ

ไม่สามารถสั่งเพิ่มอีกได้เนื่องจากเราพึ่งทำการสั่งสินค้านี้ไปและมีจำนวนสินค้าตอนนี้เป็น 0 แล้วแปลว่าสินค้าในคลังหมดเรียบร้อย

จำนวนที่สั่งพอดีกับสินค้าในคลังที่มีอยู่สามารถสั่งซื้อได้และ

รูปภาพประกอบด้วย ข้อความ, อิเล็กทรอนิกส์, ภาพหน้าจอ, ซอฟต์แวร์

คำอธิบายที่สร้างโดยอัตโนมัติ

เมื่อสั่งซื้อสินค้าเสร็จพิมพ์คำว่า end จะออกจากการสั่งซื้อและคำนวณยอดเงินสินค้าทั้งหมดที่ต้องจ่าย

แสดงตารางรายละเอียดสินค้าที่สั่งไปคือ ชื่อสินค้า , เลข id สินค้า , ราคาสินค้าต่อชิ้น , จำนวนสินค้า , ราคารวมสินค้านั้น , แบรนด์สินค้า และ หมวดหมู่สินค้า

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ตัวอักษร

คำอธิบายที่สร้างโดยอัตโนมัติ

ยอดเงินทั้งหมดที่ต้องจ่าย

จำนวนสินค้าทั้งหมดที่ซื้อ

เมื่อสั่งซื้อสินค้าเสร็จให้เปิดดูไฟล์ข้อความ 2 ไฟล์คือ

1. ไฟล์ data.txt มีการเปลี่ยนแปลงตรงจำนวนสินค้า

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, จำนวน

คำอธิบายที่สร้างโดยอัตโนมัติ

จำนวนสินค้ามีการลดลงตามจำนวนสินค้าที่ซื้อไป

1. ไฟล์ orders.txt มีการเปลี่ยนแปลงโดยจะมีข้อความอยู่ 1 ชุดที่เพิ่มเข้ามาในไฟล์นี้คือข้อมูลสรุปการสั่งซื้อนั้นเอง

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

**คำสั่งที่ 5 เพิ่มสินค้า**

พิมพ์เลข 5 เข้าไปในโปรแกรมจะมีข้อความแสดงขึ้นดังภาพต่อไปนี้

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

แสดงข้อความเพิ่มสินค้าสำเร็จ

ตั้งชื่อแบรนด์สินค้าหรือถ้าไม่มีแบรนด์ให้พิมพ์ - แปลว่าสินค้านั้นไม่มีแบรนด์

ตั้งราคาสินค้า

พิมพ์ชื่อสินค้า

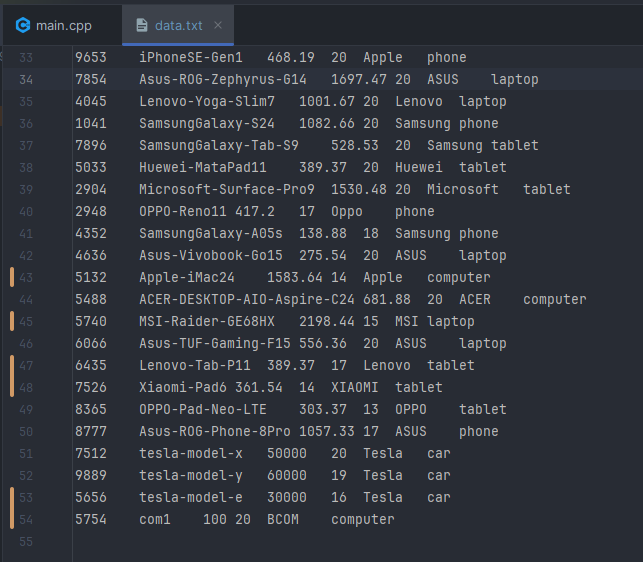
พิมพ์หมวดหมู่สินค้าที่ต้องการเพิ่ม

แสดงหมวดหมู่สินค้าที่สามารถเพิ่มสินค้าได้

**ข้อควรระวัง**

1. การแก้ไขชื่อสินค้าและชื่อแบรนด์สินค้านั้นควรเขียนชื่อให้ติดกันเป็นคำๆหรือใช้เครื่องหมาย - หรือ \_ ขั้น เพราะไม่งั้นโปรแกรมจะไม่สามารถโหลดข้อมูลสินค้าในไฟล์ data.txt ได้
2. การแก้ไขเลข id สินค้านั้นควรแก้เป็นเลขจำนวนเต็มบวกเท่านั้นและควรเป็นเลข 4 หลักเพื่อง่ายต่อการจำ

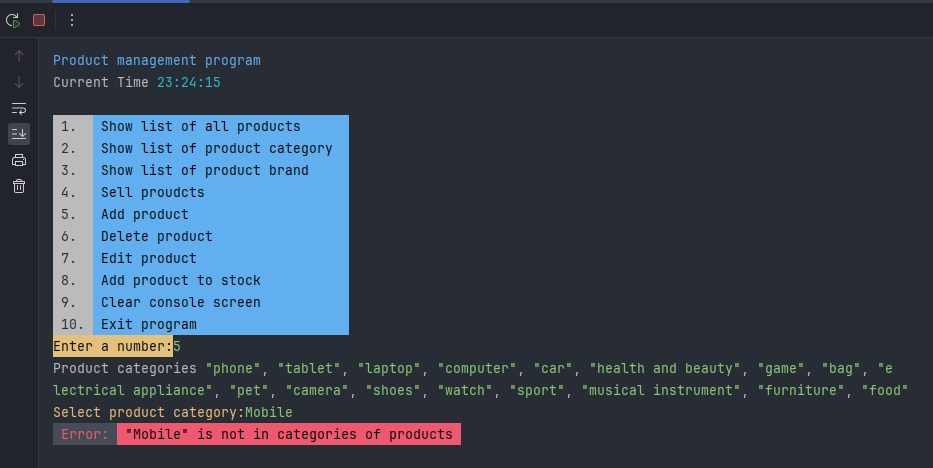
เปิดดูไฟล์ data.txt จะมีสินค้าเพิ่มขึ้นมา 1 บรรทัดคือข้อมูลที่เราเพิ่มไป



มีข้อมูลที่เพิ่มเข้ามา 1 บรรทัดโดยการเรียงข้อมูลเป็นดังนี้

(เลข id , ชื่อสินค้า , ราคาสินค้า , แบรนด์สินค้า , หมวดหมู่สินค้า) โดยจะเว้นระยะห่าง 1 tab

ข้อผิดพลาดที่สามราถเกิดขึ้นได้จากการเพิ่มสินค้าถูกรับมือไว้หมดเรียบร้อยแล้ว

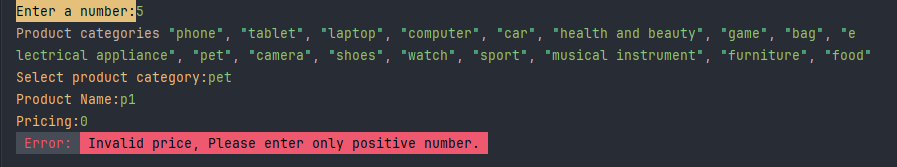


หมวดหมู่สินค้านี้ไม่มีอยู่ในหมวดหมู่สินค้าที่ระบุไว้จึงไม่สามารถทำการเพิ่มสินค้าได้และจะเริ่มที่หน้าจอเริ่มต้นของโปรแกรม

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ตัวอักษร

คำอธิบายที่สร้างโดยอัตโนมัติ

ชื่อสินค้าตัวแรกจะต้องไม่เป็นตัวเลขต้องเป็นตัวอักษรเท่านั้น



ราคาที่ตั้งต้องไม่เป็นเลขจำนวนเต็มศูนย์

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร

คำอธิบายที่สร้างโดยอัตโนมัติ

ราคาที่ตั้งต้องไม่เป็นเลขจำนวนเต็มลบ

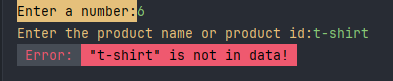
**คำสั่งที่ 6 ลบสินค้า**

พิมพ์เลข 6 เข้าไปในโปรแกรมจะมีข้อความแสดงขึ้นดังภาพต่อไปนี้

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, แสดง

คำอธิบายที่สร้างโดยอัตโนมัติ

พิมพ์ชื่อสินค้า หรือ เลข id ก็ได้เพื่อลบสินค้านี้ออกจากข้อมูล

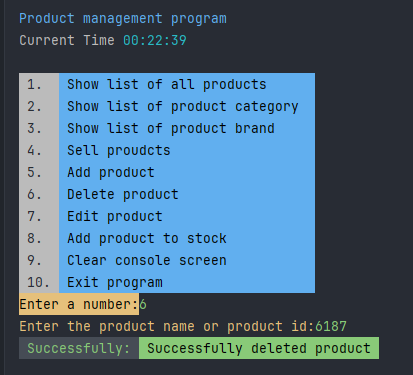
 รูปภาพประกอบด้วย ข้อความ, ตัวอักษร, ภาพหน้าจอ, จำนวน

คำอธิบายที่สร้างโดยอัตโนมัติ

ชื่อสินค้า และ เลข id ไม่มีอยู่จริงในข้อมูลที่เก็บไว้ไม่สามารถทำการสินค้าลบได้สินค้าได้

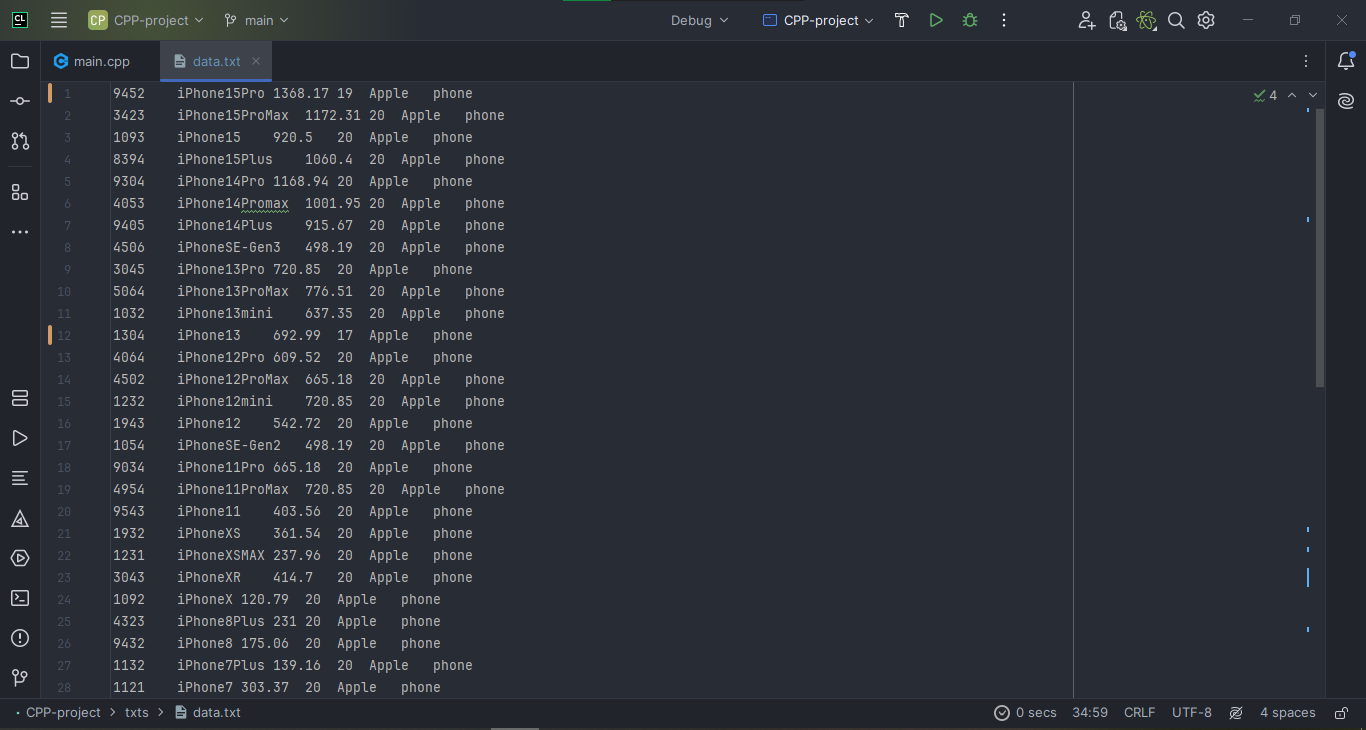
เมื่อต้องการลบสินค้าให้ใช้คำสั่งหมายเลข 1 เพื่อเล็งดูสินค้าที่ต้องการจะลบออกแล้วใช้คำสั่งหมายเลข 6 เพื่อลบสินค้านั้นออกจากโปรแกรม

รูปภาพประกอบด้วย ข้อความ, อิเล็กทรอนิกส์, ภาพหน้าจอ, ซอฟต์แวร์

คำอธิบายที่สร้างโดยอัตโนมัติ 

พิมพ์ ชื่อ หรือ เลข id ก็ได้เมื่อเช็คแล้วว่ามีข้อมูลสินค้านั้นอยู่จริงจะทำการลบสินค้านั้นออกจากโปรแกรมและแสดงข้อความว่าลบสินค้าสำเร็จ

เมื่อลบสินค้านั้นแล้วลองเข้าไปดูที่ไฟล์ data.txt จะเห็นว่ามีการลบข้อมูลสินค้าชุดนั้นลบออกไป 1 บรรทัด

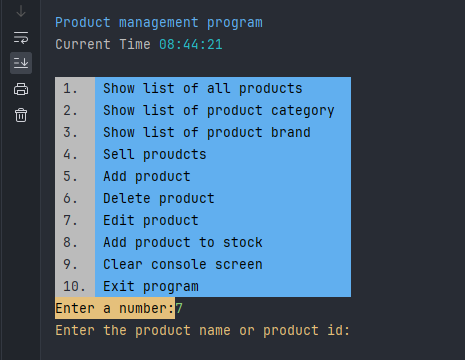


รูปภาพประกอบด้วย ภาพหน้าจอ, ข้อความ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

**คำสั่งที่ 7 แก้ไขสินค้า**

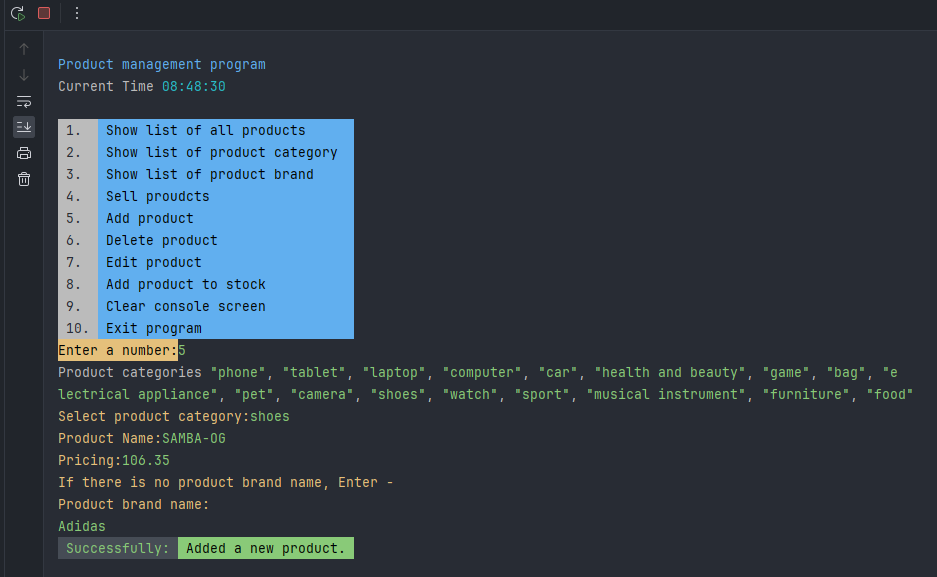
พิมพ์เลข 7 เข้าไปในโปรแกรมจะมีข้อความแสดงขึ้นดังภาพต่อไปนี้



พิมพ์ ชื่อ หรือ เลข id ก็ได้เพื่อแก้ไขสินค้า

แล้วอย่างเหมือนเดิมพิมพ์คำสั่งหมายเลข 1 เพื่อดูสินค้าที่จะแก้ไข และ คำสั่งหมายเลข 7 เพื่อแก้ไขสินค้านั้น

ในตัวอย่างนี้จะเพิ่มสินค้า 1 ชิ้นก่อนแล้วทำการแก้ไขสินค้าตัวนั้น



เพิ่มสินค้าเพื่อทดลองแก้ไขสินค้า

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, แสดง

คำอธิบายที่สร้างโดยอัตโนมัติ

มีสินค้าเพิ่มเข้ามาใหม่ในตาราง

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, แสดง

คำอธิบายที่สร้างโดยอัตโนมัติ

พิมพ์ชื่อสินค้า หรือ เลข id ก็ได้แต่ตัวอย่างนี้จะใช้เป็นชื่อสินค้าแทน

แก้ไขข้อมูลสินค้าอันเก่าโดยแก้ไข ชื่อสินค้า , ราคาสินค้า และ แบรนด์สินค้า

คำถามว่าจะแก้ไขสินค้านั้นในส่วนอะไรบ้างพิมพ์ตอบได้แค่ y และ n เท่านั้น ถ้าพิมพ์ y จะมีข้อความขึ้นมาแล้วให้เราพิมพ์ข้อมูลใหม่ลงไปเพื่อแก้ไขข้อมูลสิน ถ้าพิมพ์ n จะแสดงคำถามตัวต่อไป

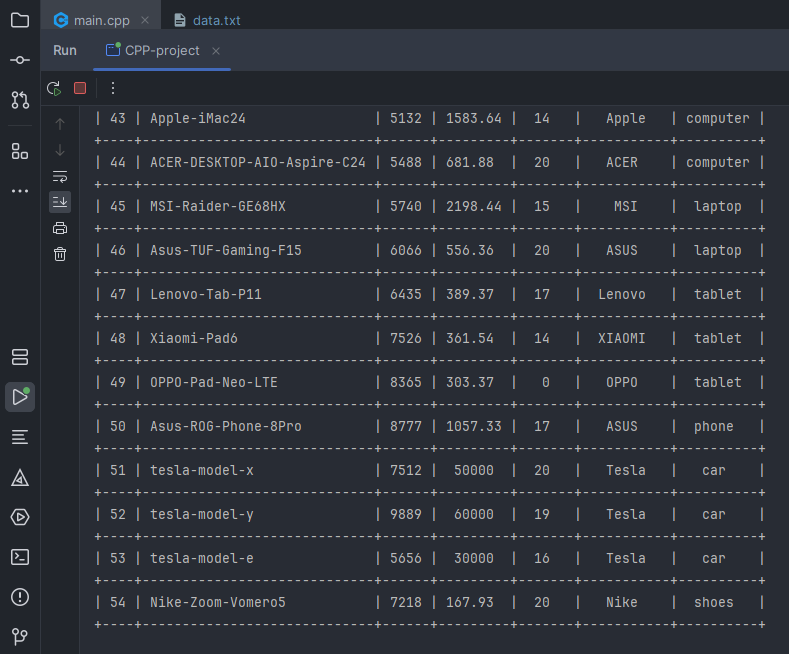
แสดงรายละเอียดสินค้าตัวนั้นเพื่อจะแก้ไขในแต่ละส่วน

เข้าไปดูไฟล์ data.txt และพิมพ์คำสั่งหมายเลข 1 เพื่อดูว่าข้อมูลที่แก้ไขนั้นมีการเปลี่ยนแปลงจริงหรือไม่

รูปภาพประกอบด้วย ภาพหน้าจอ, ข้อความ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

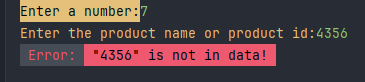
คำอธิบายที่สร้างโดยอัตโนมัติ

จะเห็นว่าข้อมูลบรรทัดที่ 54 มีการเปลี่ยนแปลงไปตามข้อมูลที่เราแก้ไข



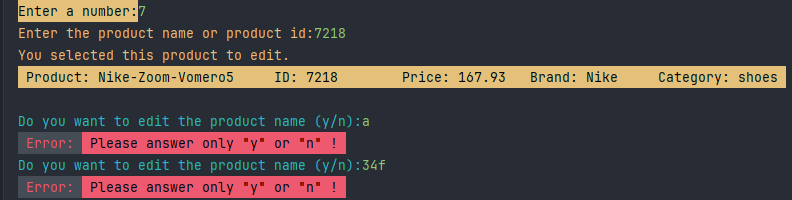
ข้อมูลเปลี่ยนแปลงไป

ข้อผิดพลาดที่สามราถเกิดขึ้นได้จากการแก้ไขสินค้าถูกรับมือไว้หมดเรียบร้อยแล้ว

 รูปภาพประกอบด้วย ข้อความ, ตัวอักษร, ภาพหน้าจอ, ไลน์

คำอธิบายที่สร้างโดยอัตโนมัติ

เช็คสินค้าที่จะแก้ไขนั้นไม่มีอยู่จริงกับข้อมูลที่เก็บจะไม่ดำเนินการแก้ไขใดๆ



พิมพ์ตอบได้แค่ y กับ n ได้เท่านั้นไม่งั้นจะแสดงข้อความเดิม

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, จำนวน

คำอธิบายที่สร้างโดยอัตโนมัติ

ชื่อสินค้าที่แก้ไขไม่สามารถตั้งชื่ออักษรตัวแรกเป็นตัวเลขได้

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

ชื่อสินค้าและเลข id ไม่สามาถใส่ซ้ำกับข้อมูลสินค้าที่มีอยู่ได้

**ข้อควรระวัง**

1. การแก้ไขชื่อสินค้าและชื่อแบรนด์สินค้านั้นควรเขียนชื่อให้ติดกันเป็นคำๆหรือใช้เครื่องหมาย - หรือ \_ ขั้น เพราะไม่งั้นโปรแกรมจะไม่สามารถโหลดข้อมูลสินค้าในไฟล์ data.txt ได้
2. การแก้ไขเลข id สินค้านั้นควรแก้เป็นเลขจำนวนเต็มบวกเท่านั้นและควรเป็นเลข 4 หลักเพื่อง่ายต่อการจำ

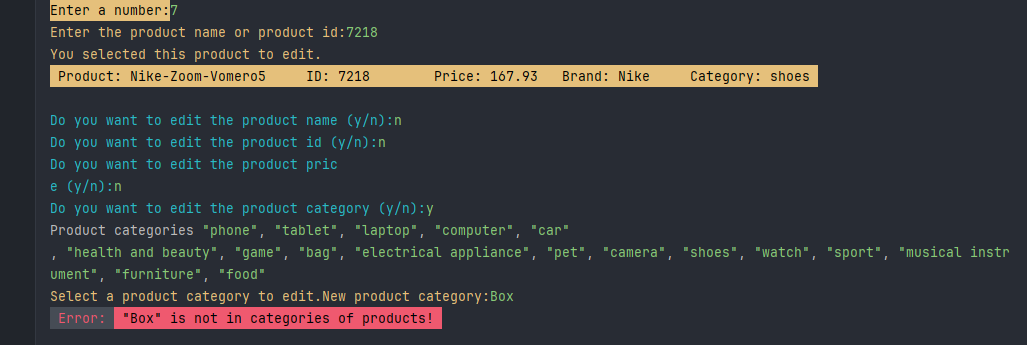
รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, จำนวน

คำอธิบายที่สร้างโดยอัตโนมัติ

การแก้ไขราคาสินค้าต้องตั้งเป็นเลขจำนวนเต็มบวกเท่านั้น

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, จำนวน

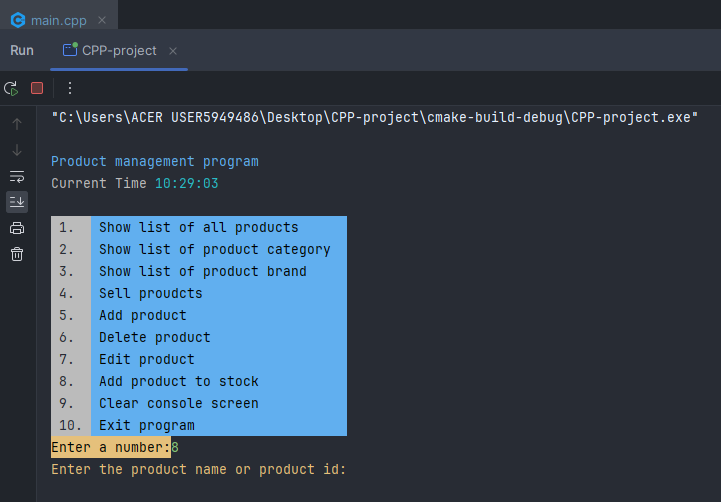
คำอธิบายที่สร้างโดยอัตโนมัติ



หมวดหมู่สินค้าที่แก้ไขต้องต้องอยู่ในหมวดหมู่เดียวกับที่ระบุไว้ในภาพ

**คำสั่งที่ 8 เพิ่มสินค้าใน stock**

พิมพ์เลข 8 เข้าไปในโปรแกรมจะมีข้อความแสดงขึ้นดังภาพต่อไปนี้



ใส่ชื่อสินค้า หรือ เลข id เพื่อค้นหาสินค้า

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

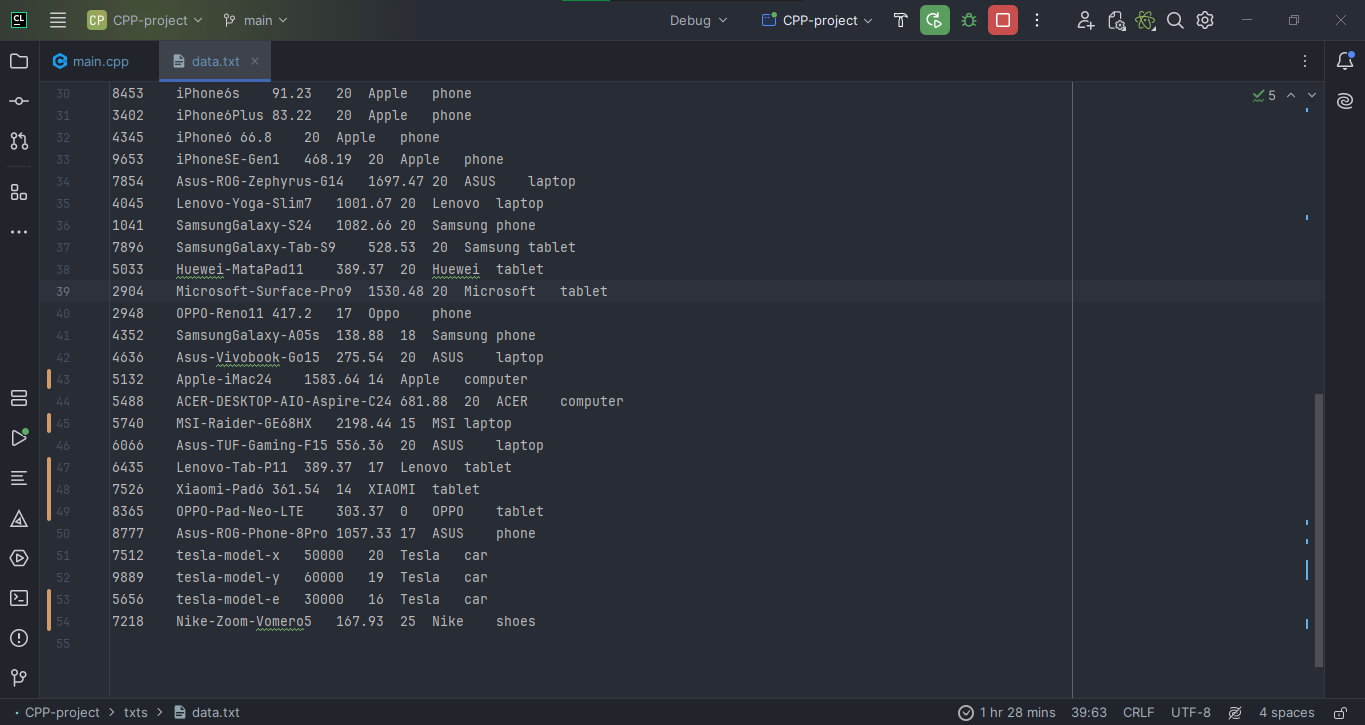
คำอธิบายที่สร้างโดยอัตโนมัติ

และเหมือนเดิมใช้คำสั่งหมายเลข 1 และ 8 ร่วมกัน

จำนวนสินค้าที่จะเพิ่มเข้าไปใน stock (จำนวนทีใส่เข้าไปเราไม่ได้จำกัดจะใส่จำนวนเยอะเท่าไหร่ก็ได้แต่ควรใส่จำนวนให้เป็นความจริง)

แสดงข้อความเพิ่มจำนวนสินค้าเข้าไปใน stock สำเร็จ

ลองสังเกตดุที่ไฟล์ data.txt ข้อมูลนั้นจะมีจำนวนสินค้าใน stock เพิ่มขึ้นจากเดิม



ข้อมูลบรรทัดที่ 54 มีจำนวนเพิ่มขึ้น

ใช้คำสั่งหมายเลข 1 เพื่อดูว่าจำนวนสินค้าใน stock ตัวนี้เพิ่มขึ้นจริงไหม

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

มีจำนวนเพิ่มขึ้นจริงตามไฟล์ data.txt

รูปภาพประกอบด้วย ข้อความ, ตัวอักษร, ภาพหน้าจอ, ไลน์

คำอธิบายที่สร้างโดยอัตโนมัติ รูปภาพประกอบด้วย ข้อความ, ตัวอักษร, ภาพหน้าจอ, กราฟิก

คำอธิบายที่สร้างโดยอัตโนมัติ

ไม่มีชื่อสินค้า และ เลข id นี้อยู่ในข้อมูลแสดงข้อความ error ดังภาพ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, ไลน์

คำอธิบายที่สร้างโดยอัตโนมัติ

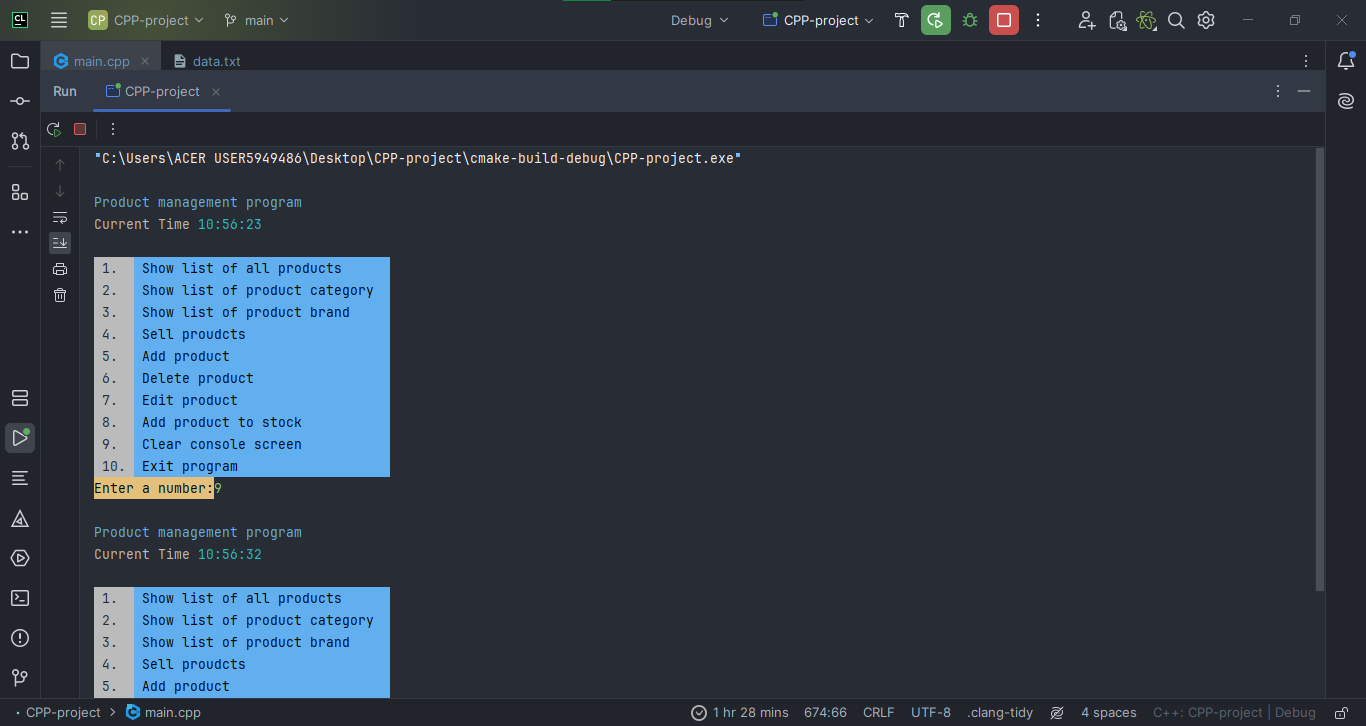
จำนวนที่เพิ่มเข้าไปต้องเป็นเลขจำนวนเต็มบวกเท่านั้น

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, จำนวน

คำอธิบายที่สร้างโดยอัตโนมัติ

**คำสั่งที่ 9 ล้างหน้าจอโปรแกรม**

พิมพ์เลข 9 เข้าไปในโปรแกรมตัวโปรแกรมจะล้างหน้าจอ console ให้



แต่เนื่องจากผู้เขียนโปรแกรมนี้พัฒนาโดยใช้ IDE เป็น CLion เมื่อพิมพ์เลข 9 เข้าไปแล้วจะไม่เห็นการเปลี่ยนแปลงในหน้าจอ console คำสั่งนี้จึงสามารถใช้ได้กับ โปรแกรม .exe

เปิดโปรแกรม CPP-project.exe ที่อยู่ในโฟลเดอร์ของโปรเจค ลองพิมพ์คำสั่งเล่นๆเพื่อให้หน้าจอเต็มไปด้วยข้อความแล้วใช้คำสั่งหมายเลข 9 เพื่อล้างหน้าจอ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ภาพหน้าจอ, ข้อความ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย, แสดง

คำอธิบายที่สร้างโดยอัตโนมัติ

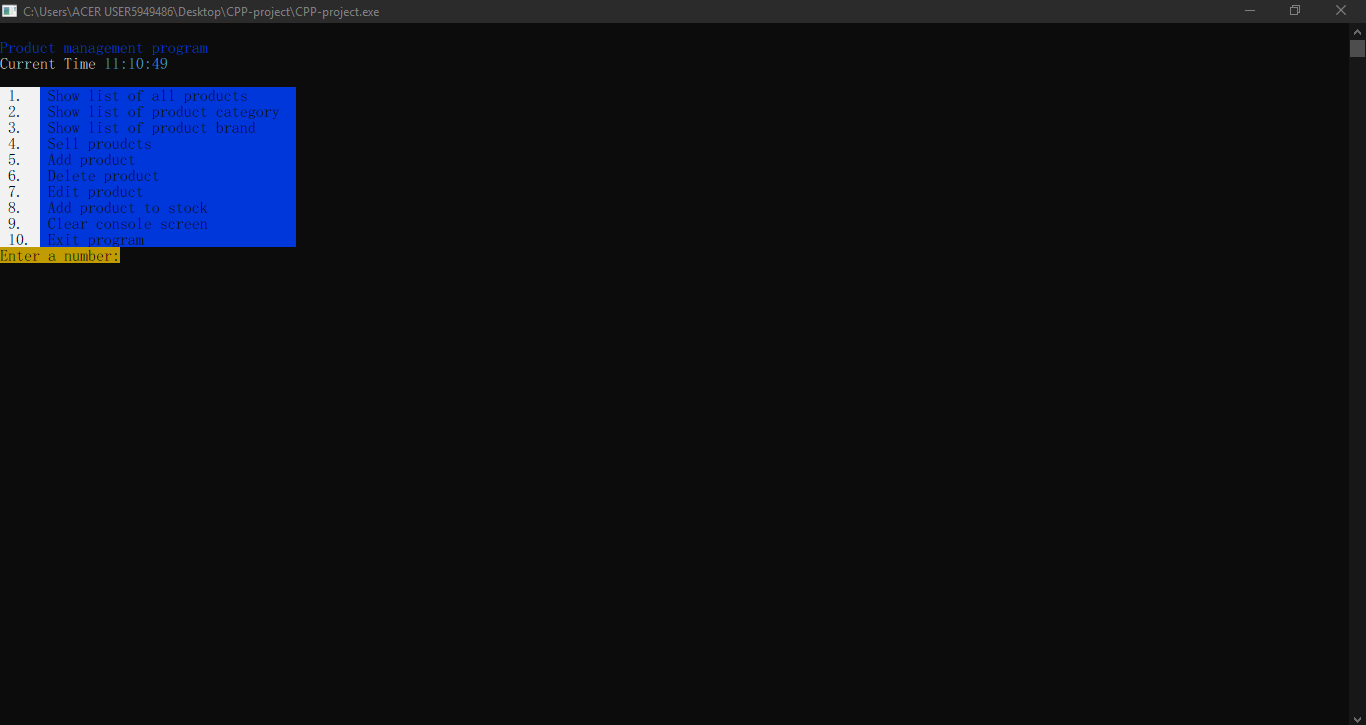
รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, แสดง

คำอธิบายที่สร้างโดยอัตโนมัติ

พิมพ์เลข 9 เพื่อล้างหน้าจอ



หน้าจอถูกล้างเสร็จเรียบร้อย

**คำสั่งที่ 10 ออกจากโปรแกรม**

พิมพ์เลข 10 เข้าไปในโปรแกรมโปรแกรมจะจบการทำงานลง

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ซอฟต์แวร์มัลติมีเดีย

คำอธิบายที่สร้างโดยอัตโนมัติ

**แหล่งเอกสารอ้างอิงและการใช้งานคำสั่ง APIs ทั้งหมดของ Libraries**

1. https://marcuscode.com/lang/cpp/files
2. https://medium.com/@vachirachat.saw/%E0%B9%82%E0%B8%84%E0%B8%A3%E0%B8%87%E0%B8%AA%E0%B8%A3%E0%B9%89%E0%B8%B2%E0%B8%87%E0%B8%82%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%B9%E0%B8%A5%E0%B8%95%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%86%E0%B9%83%E0%B8%99-c-1ccdb706c38e
3. https://www.devdit.com/post/4200/c-plus-plus-vector-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B8%97%E0%B8%B3%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%A2%E0%B8%B1%E0%B8%87%E0%B9%84%E0%B8%87#gsc.tab=0
4. https://medium.com/@marktbss/c-hackerrank-vector-erase-11c65b830a43
5. https://www.geeksforgeeks.org/how-to-clear-console-in-cpp
6. https://www.javatpoint.com/cpp-date-and-time
7. https://github.com/seleznevae/libfort
8. https://seleznevae.github.io/libfort/index.html
9. https://cplusplus.com/reference/cstdlib/rand
10. https://github.com/ikalnytskyi/termcolor
11. https://termcolor.readthedocs.io/#
12. https://weerasak.dev/posts/2023/03/18/basic-cmake-for-building-c-cpp-project/