| ***Computer Engineering Department*** |
| --- |
| ***CE100L: Computing Fundamentals & Programming*** |

| ***Course Instructor: Usama Bin Shakeel*** | ***Dated: 21/01/2022*** |
| --- | --- |
| ***Teaching Assistant: Aqsa Khalid*** | ***Semester: Fall 2021*** |
| ***Lab Engineer: Nadir Abbas*** | ***Batch: BSCE2021*** |

# **Lab 13B. Challenging Problem Practice in C++**

| **Name** | **Roll number** | **Report**  **(out of 100)** | **Scaled to 10** | **Total**  **(out of 10)** |
| --- | --- | --- | --- | --- |
| NIMRA MAQBOOL | BSCE21012 |  |  |  |

Checked on: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Objective**

The objective of this lab is to understand open ended problem solving.

## **Equipment and Component**

| **Component Description** | **Value** | **Quantity** |
| --- | --- | --- |
| Computer | Available in lab | 1 |

## **Conduct of Lab**

1. Students are required to perform this experiment individually.
2. In case the lab experiment is not understood, the students are advised to seek help from the course instructor, lab engineers, assigned teaching assistants (TA) and lab attendants.

## **Theory and Background**

A challenge is something new and difficult which requires great effort and determination.Every problem has a solution, we just need to think and try to solve it. We can solve every problem, we just need some courage to solve it. It depends on the problem, how much courage we require to solve that problem.



Figure 1: \*What is Open Ended Problem Solving??

\*https://medium.com/@aitisam.zafeer88/problem-solving-challenge-322a3ee7db23

**Lab Task**

1. A customer wants you to design and then build an online store. While interviewing the customer following information was extracted:

Track products and related information.   
To do: extract details (i.e. other information to track) from interview notes.

Abilities the customer wants in the system:

Identify a product that will be for sale. You should read in the name, model number, wholesale cost, and retail cost for the product.Enter a new customer. You should read in the customer name and ID.

Take a shipment of new products. Read in the model number and quantity. If you don't know what the product is that you're getting, reject the shipment, otherwise add that to inventory.

Let a customer buy something. The customer ID, product model number, and quantity should be taken as input. If there is sufficient quantity of the product on hand, then the customer should be charged that amount and the product be deducted from inventory. If there is not sufficient quantity, the sale should be rejected. Let a customer make a payment. Read in the customer ID and the amount of payment. It's OK for customers to have a positive balance, but they cannot make negative payments. Find out about a customer: enter a customer ID number, and print out the customer's name, current balance, and a list of what the customer has previously purchased. Find out about a product: enter a model number and get the name of the product, the amount that has already been sold, and the amount in inventory. Print lists of all information about all customers and all products.

| struct product {  char name[50];  int wholeSale;  float retailPrice;  int modelNum;  float tax;  string customerName;  float balance;  int ordersPrevious;  int quantity;  string pass;  string id;  float sum;  int price;  };  void loginSystem() {  product p1;  ofstream myFile;  myFile.open("loginForAdmin.txt", ios::app);  if (myFile.is\_open()) {  cout << "please enter the id of admin." << endl;  cin >> p1.id;  cout << "please enter a strong password." << endl;  cin.get();  getline(cin, p1.pass);  myFile << "The id you entered is " << p1.id << endl;  myFile << "the pass you entered is " << p1.pass << endl;  cout << "YOUR ACCOUNT HAS BEEN CREATED 0\_0" << endl;  } else {  cout << "The file you are trying to open does not exit -\_-" << endl;  }  myFile.close();  }  void shipment() {  int numberOfProducts;  product p1;  cout << "please enter the model number you want to add to the inventory." << endl;  cin >> p1.modelNum;  fstream fileX;  if (p1.modelNum != 431) {  cout << "there is no such model number,So I can't take your shipment order.";  } else if (p1.modelNum == 431) {  fileX.open("modelXForShipping.txt", ios::in | ios::out | ios::app);  if (fileX.is\_open()) {  cout << ".............. FOR MODEL 431 ................." << endl;  cout << "please enter number of products you need." << endl;  cin >> numberOfProducts;  for (int i = 1; i <= numberOfProducts; i++) {  cout << "\nEnter product name." << endl;  cin >> p1.name;  // cout << "Enter whole sale price" << endl;  // cin >> p1.wholeSale;  // cout << "Enter retail price" << endl;  // cin >> p1.retailPrice;  cout << "\nThe name Of the Product = " << setw(15) << p1.name << setw(30) << endl;  // "wholeSale of the product = " << setw(15) << p1.wholeSale << setw(30)  // << "retail price of the product = " << setw(15) << p1.retailPrice << endl;  fileX << "\nThe name Of the Product = " << setw(15) << p1.name << setw(30) << endl;  // << "wholeSale of the product = " << setw(15) << p1.wholeSale << setw(30)  // << "retail price of the product = " << setw(15) << p1.retailPrice << endl;  }  }  cout << "your products have been added to the inventory..." << endl;  fileX << "your products have been added to the inventory..." << endl;  fileX.close();  }  }  void shipmentModel432() {  int numberOfProducts;  product p1;  cout << "please enter the model number you want to add to the inventory." << endl;  cin >> p1.modelNum;  fstream fileY;  if (p1.modelNum != 432) {  cout << "there is no such model number,so I can't take your shipment order .";  } else if (p1.modelNum == 432) {  fileY.open("modelYForShipping.txt", ios::in | ios::out | ios::app);  if (fileY.is\_open()) {  cout << "================= FOR MODEL 432 =================" << endl;  cout << "please enter number of products you need." << endl;  cin >> numberOfProducts;  for (int i = 1; i <= numberOfProducts; i++) {  cout << endl;  cout << "\nplease enter product name." << endl;  cin >> p1.name;  // cout << "please enter whole sale price" << endl;  // cin >> p1.wholeSale;  // cout << "please enter retail price" << endl;  // cin >> p1.retailPrice;  cout << " The name Of the Product = " << setw(15) << p1.name << setw(30) << endl;  // << "wholeSale of the product = " << setw(15) << p1.wholeSale << setw(30)  // << "retail price of the product = " << setw(15) << p1.retailPrice << endl;  fileY << " The name Of the Product = " << setw(15) << p1.name << setw(30) << endl;  // << "wholeSale of the product = " << setw(15) << p1.wholeSale << setw(30)  // << "retail price of the product = " << setw(15) << p1.retailPrice << endl;  cout << endl;  }  cout << "your products have been added to the inventory..." << endl;  cout << endl;  fileY << "your products have been added to the inventory..." << endl;  cout << endl;  }  fileY.close();  } else {  cout << "The file you are trying to open does not exit -\_-" << endl;  }  }  void shipmentModel433() {  int numberOfProducts;  product p1;  cout << "please enter the model number you want to add to the inventory." << endl;  cin >> p1.modelNum;  fstream fileZ;  if (p1.modelNum != 433) {  cout << "there is no such model number,so I can't take your shipment order.";  } else if (p1.modelNum == 433) {  fileZ.open("modelZForShipping.txt", ios::in | ios::out | ios::app);  if (fileZ.is\_open()) {  cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ FOR MODEL 433 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;  cout << "please enter number of products you need." << endl;  cin >> numberOfProducts;  for (int i = 0; i < numberOfProducts; i++) {  cout << "\nplease enter product name." << endl;  cin >> p1.name;  // cout << "please enter whole sale price" << endl;  // cin >> p1.wholeSale;  // cout << "please enter retail price" << endl;  // cin >> p1.retailPrice;  cout << " The name Of the Product = " << setw(15) << p1.name << setw(30) << endl;  // << "wholeSale of the product = " << setw(15) << p1.wholeSale << setw(30)  // << "retail price of the product = " << setw(15) << p1.retailPrice << endl;  fileZ << " The name Of the Product = " << setw(15) << p1.name << setw(30) << endl;  // "wholeSale of the product = " << setw(15) << p1.wholeSale << setw(30)  // << "retail price of the product = " << setw(15) << p1.retailPrice << endl;  }  }  cout << "your products have been added to the inventory..." << endl;  cout << endl;  fileZ << "your products have been added to the inventory..." << endl;  cout << endl;  fileZ.close();  } else {  cout << "The file you are trying to open does not exit -\_-" << endl;  }  }  void menuForShipment() {  int opt;  do {  cout << "---------------------- choose options for shipment ---------------------" << endl;  cout << "1.For shipment of model 431." << endl;  cout << "2.For shipment of model 432." << endl;  cout << "3.For shipment of model 433." << endl;  cout << "4.exit." << endl;  cin >> opt;  switch (opt) {  case 1: {  shipment();  break;  }  case 2: {  shipmentModel432();  break;  }  case 3: {  shipmentModel433();  break;  }  case 4: {  cout << "you choose to exit." << endl;  exit(1);  break;  }  default: {  cout << "you have entered invalid option.";  }  }  } while (opt >= 1 && opt <= 4);  }  void productA() {  product p1;  int numberOfProducts1;  cout << "please enter the product model you want to add to the inventory 0\_0.. " << endl;  cin >> p1.modelNum;  fstream file;  if (p1.modelNum == 431) {  file.open("modelA.txt", ios::in | ios::out | ios::app);  if (file.is\_open()) {  cout << "please enter number of products you need..." << endl;  cin >> numberOfProducts1;  if(numberOfProducts1>10 ){  cout<<"we cant add more than 10 items in inventory..";  }  else {  for (int i = 1; i <= numberOfProducts1; i++) {  cout << "please enter product name." << endl;  cin >> p1.name;  cout << "please enter whole sale price" << endl;  cin >> p1.wholeSale;  cout << "please enter price" << endl;  cin >> p1.price;  if (p1.price <= 1000 && p1.price > 0) {  cout << "a tax of 2% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.02));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax << endl;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else if (p1.price <= 2000 && p1.price > 1000) {  cout << "a tax of 3% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.03));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else if (p1.price <= 3000 && p1.price > 2000) {  cout << "a tax of 4% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.04));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else if (p1.price <= 4000 && p1.price > 3000) {  cout << "a tax of 5% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.05));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else {  cout << "a tax of 10% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.1));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  }  cout << "name =" << setw(10) << p1.name << setw(20) << "wholeSale =" << setw(10) << p1.wholeSale  << setw(20) << " price =" << setw(10) << p1.price << setw(20) << "tax =" << setw(10) << p1.tax  << setw(20) << "retail price = " << setw(10) << p1.retailPrice << endl;  file << "name =" << setw(10) << p1.name << setw(20) << "wholeSale =" << setw(10) << p1.wholeSale  << setw(20) << " price =" << setw(10) << p1.price << setw(10) << "tax =" << setw(10) << p1.tax  << setw(20) << "retail price =" << setw(10) << p1.retailPrice << endl;  }  }  }  file.close();  } else {  cout << "you have entered invalid model number." << endl;  }  }  void productB() {  product p1;  cout << "please enter the product model you want to add to the inventory 0\_0.. " << endl;  cin >> p1.modelNum;  int numberOfProductsB;  fstream file1;  if (p1.modelNum == 432) {  file1.open("modelB.txt", ios::in | ios::out | ios::app);  if (file1.is\_open()) {  cout << "please enter number of products you need..." << endl;  cin >> numberOfProductsB;  if(numberOfProductsB>10 ){  cout<<"we cant add more than 10 items in inventory.."<<endl;  }  else {  for (int i = 1; i <= numberOfProductsB; i++) {  cout << "please enter product name." << endl;  cin >> p1.name;  cout << "please enter whole sale price" << endl;  cin >> p1.wholeSale;  cout << "please enter price" << endl;  cin >> p1.price;  if (p1.price <= 1000 && p1.price > 0) {  cout << "a tax of 2% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.02));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax << endl;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else if (p1.price <= 2000 && p1.price > 1000) {  cout << "a tax of 3% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.03));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else if (p1.price <= 3000 && p1.price > 2000) {  cout << "a tax of 4% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.04));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else if (p1.price <= 4000 && p1.price > 3000) {  cout << "a tax of 5% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.05));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else {  cout << "a tax of 10% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.1));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  }  cout << "name =" << setw(10) << p1.name << setw(20) << "wholeSale =" << setw(10) << p1.wholeSale  << setw(20) << " price =" << setw(10) << p1.price << setw(10) << "tax =" << setw(10) << p1.tax  << setw(20) << "retail price = " << setw(10) << p1.retailPrice << endl;  file1 << "name =" << setw(10) << p1.name << setw(20) << "wholeSale =" << setw(10) << p1.wholeSale  << setw(20) << " price =" << setw(10) << p1.price << setw(10) << "tax =" << setw(10) << p1.tax  << setw(20) << "retail price =" << setw(10) << p1.retailPrice << endl;  }  }  }  file1.close();  } else {  cout << "you have entered invalid model number." << endl;  }  }  void productC() {  product p1;  int numberOfProductsC;  cout << "please enter the product model you want to add to the inventory 0\_0.. " << endl;  cin >> p1.modelNum;  fstream file2;  if (p1.modelNum == 433) {  file2.open("modelC.txt", ios::in | ios::out | ios::app);  if (file2.is\_open()) {  cout << "please enter number of products you need..." << endl;  cin >> numberOfProductsC;  if(numberOfProductsC>10 ){  cout<<"we cant add more than 10 items in inventory.."<<endl;  }  else {  for (int i = 1; i <= numberOfProductsC; i++) {  cout << "please enter product name." << endl;  cin >> p1.name;  cout << "please enter whole sale price" << endl;  cin >> p1.wholeSale;  cout << "please enter price" << endl;  cin >> p1.price;  if (p1.price <= 1000 && p1.price > 0) {  cout << "a tax of 2% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.02));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax << endl;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else if (p1.price <= 2000 && p1.price > 1000) {  cout << "a tax of 3% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.03));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else if (p1.price <= 3000 && p1.price > 2000) {  cout << "a tax of 4% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.04));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else if (p1.price <= 4000 && p1.price > 3000) {  cout << "a tax of 5% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.05));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  } else {  cout << "a tax of 10% is applied on retail price (｡◕‿◕｡)" << endl;  p1.tax = ((p1.price) \* (0.1));  p1.tax = (p1.tax / p1.wholeSale);  cout << "the tax applied is = " << setw(10) << p1.tax;  p1.retailPrice = p1.price + p1.tax;  cout << "the total Price is = " << p1.retailPrice;  }  cout << "name =" << setw(10) << p1.name << setw(20) << "wholeSale =" << setw(10) << p1.wholeSale  << setw(20) << " price =" << setw(10) << p1.price << setw(10) << "tax =" << setw(10) << p1.tax  << setw(20) << "retail price =" << setw(10) << p1.retailPrice << endl;  file2 << "name =" << setw(10) << p1.name << setw(20) << "wholeSale =" << setw(10) << p1.wholeSale  << setw(20) << " price =" << setw(10) << p1.price << setw(10) << "tax =" << setw(10) << p1.tax  << setw(20) << "retail price =" << setw(10) << p1.retailPrice << endl;  }  }  }  file2.close();  }  }  void menuForNewProduct() {  int opt1;  do {  cout << "-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_- choose new product -\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-" << endl;  cout << "1.For product of model 431." << endl;  cout << "2.For product of model 432." << endl;  cout << "3.For product of model 433." << endl;  cout << "4.exit." << endl;  cin >> opt1;  switch (opt1) {  case 1: {  productA();  break;  }  case 2: {  productB();  break;  }  case 3: {  productC();  break;  }  case 4: {  cout << "you choose to exit." << endl;  exit(1);  break;  }  default: {  cout << "you have entered invalid option.";  }  }  } while (opt1 >= 1 && opt1 <= 4);  }  void displayModelA() {  product p1;  fstream file;  file.open("modelA.txt", ios::ate | ios::in);  string data;  if (file.is\_open()) {  if (!file.tellg() == 0) {  file.seekg(0, ios::beg);  while (!file.eof()) {  getline(file, data);  cout << data << endl;  }  } else {  cout << "there is nothing in this file.-\_-." << endl;  }  } else {  cout << "your file isn't open -\_-" << endl;  }  file.close();  }  void displayModelB() {  product p1;  fstream file1;  file1.open("modelB.txt", ios::ate | ios::in);  string data1;  if (file1.is\_open()) {  if (!file1.tellg() == 0) {  file1.seekg(0, ios::beg);  while (!file1.eof()) {  getline(file1, data1);  cout << data1 << endl;  }  } else {  cout << "there is nothing in this file.-\_-." << endl;  }  } else {  cout << "your file isn't open -\_-" << endl;  }  file1.close();  }  void displayModelC() {  product p1;  fstream file2;  file2.open("modelC.txt", ios::ate | ios::in);  string data2;  if (file2.is\_open()) {  if (!file2.tellg() == 0) {  file2.seekg(0, ios::beg);  while (!file2.eof()) {  getline(file2, data2);  cout << data2 << endl;  }  } else {  cout << "there is nothing in this file.-\_-." << endl;  }  } else {  cout << "your file isn't open -\_-" << endl;  }  file2.close();  }  void displayRecordMenu() {  int opt2;  do {  cout << "================== choose options for displaying record ==================" << endl;  cout << "1.For record of model 431." << endl;  cout << "2.For record of model 432." << endl;  cout << "3.For record of model 433." << endl;  cout << "4.exit." << endl;  cin >> opt2;  switch (opt2) {  case 1: {  displayModelA();  break;  }  case 2: {  displayModelB();  break;  }  case 3: {  displayModelC();  break;  }  case 4: {  cout << "you choose to exit." << endl;  exit(1);  break;  }  default: {  cout << "you have entered invalid option.";  }  }  } while (opt2 >= 1 && opt2 <= 4);  }  void shippingModelA() {  product p1;  fstream fileX;  fileX.open("modelXForShipping.txt", ios::ate | ios::in);  string dataX;  if (fileX.is\_open()) {  if (!fileX.tellg() == 0) {  fileX.seekg(0, ios::beg);  while (!fileX.eof()) {  getline(fileX, dataX);  cout << dataX << endl;  }  } else {  cout << "there is nothing in this file.-\_-." << endl;  }  } else {  cout << "your file isn't open -\_-" << endl;  }  fileX.close();  }  void shippingModelB() {  product p1;  fstream fileY;  fileY.open("modelYForShipping.txt", ios::ate | ios::in);  string dataY;  if (fileY.is\_open()) {  if (!fileY.tellg() == 0) {  fileY.seekg(0, ios::beg);  while (!fileY.eof()) {  getline(fileY, dataY);  cout << dataY << endl;  }  } else {  cout << "there is nothing in this file.-\_-." << endl;  }  } else {  cout << "your file isn't open -\_-" << endl;  }  fileY.close();  }  void shippingModelC() {  product p1;  fstream fileZ;  fileZ.open("modelZForShipping.txt", ios::ate | ios::in);  string dataZ;  if (fileZ.is\_open()) {  if (!fileZ.tellg() == 0) {  fileZ.seekg(0, ios::beg);  while (!fileZ.eof()) {  getline(fileZ, dataZ);  cout << dataZ << endl;  }  } else {  cout << "there is nothing in this file.-\_-." << endl;  }  } else {  cout << "your file isn't open -\_-" << endl;  }  fileZ.close();  }  void displayShippingRecordMenu() {  int opt4;  do {  cout << "================== choose options for displaying record ==================" << endl;  cout << "1.For record of shipping model 431." << endl;  cout << "2.For record of shipping model 432." << endl;  cout << "3.For record of shipping model 433." << endl;  cout << "4.exit." << endl;  cin >> opt4;  switch (opt4) {  case 1: {  shippingModelA();  break;  }  case 2: {  shippingModelB();  break;  }  case 3: {  shippingModelC();  break;  }  case 4: {  cout << "you choose to exit." << endl;  exit(1);  break;  }  default: {  cout << "you have entered invalid option.";  }  }  } while (opt4 >= 1 && opt4 <= 4);  }  void adminMenu() {  int opt3;  do {  cout << "--------------choose options as a admin ----------------" << endl;  cout << "1.For shipment." << endl;  cout << "2.For new product." << endl;  cout << "3.For record displaying." << endl;  cout << "4.For record displaying of shipments." << endl;  cout << "5..exit." << endl;  cin >> opt3;  switch (opt3) {  case 1: {  menuForShipment();  break;  }  case 2: {  menuForNewProduct();  break;  }  case 3: {  displayRecordMenu();  break;  }  case 4: {  displayShippingRecordMenu();  break;  }  case 5: {  cout << "you choose to exit." << endl;  exit(1);  }  default: {  cout << "you have entered invalid option.";  }  }  } while (opt3 >= 1 && opt3 <= 4);  }  void loginSystemForCustomer() {  product p1;  ofstream myFile1;  myFile1.open("loginForCustomer.txt", ios::app);  if (myFile1.is\_open()) {  cout << "please enter the id..." << endl;  cin >> p1.id;  cout << "please enter a strong password." << endl;  cin.get();  getline(cin, p1.pass);  cout << "please enter your beautiful name •ᴥ•" << endl;  cin.get();  getline(cin, p1.customerName);  cout << "please enter your account balance •ᴥ• " << endl;  cin >> p1.balance;  myFile1 << "The id you entered is = " << setw(10) << p1.id << endl;  myFile1 << "The pass you entered is= " << setw(10) << p1.pass << endl;  myFile1 << "Your name = " << setw(15) << p1.customerName << endl;  myFile1 << "Current balance in account = " << setw(20) << p1.balance << endl;  if (p1.balance < 0) {  cout << "you can't have negative amount in balance (ง’̀-‘́)ง" << endl;  } else {  cout << "YOUR ACCOUNT HAS BEEN CREATED 0\_0" << endl;  cout << "Now you can explore our website and can buy anything you want..0\_0" << endl;  }  } else {  cout << "The file you are trying to open does not exit -\_-" << endl;  }  myFile1.close();  }  void buyingModelA() {  int quantity1=10;  product p1;  fstream file;  cout << "please tell us the model of the product you want to buy from ten inventory products.." << endl;  cin >> p1.modelNum;  if (p1.modelNum == 431) {  if (quantity1 <= 0) {  cout << "Their is nothing in the inventory (ᵔᴥᵔ)" << endl;  exit(1);  }  cout << "How much quantity of products you want to buy.." << endl;  cin >> p1.quantity;  if (p1.quantity <= quantity1) {  cout << "please enter product name." << endl;  cin >> p1.name;  file.open("modelA.txt", ios::ate | ios::in);  string data1;  if (file.is\_open()) {  cout<<"==============you have bought the following things============="<<endl<<endl;  if(p1.quantity==1){  file.seekg((p1.quantity+10)\*104,ios::beg);  while (getline(file, data1)) {  cout << data1 << endl;  }  }  else if(p1.quantity==2){  file.seekg((p1.quantity+9)\*92.5,ios::beg);  while (getline(file, data1)) {  cout << data1 << endl;  }  }  else if(p1.quantity==3){  file.seekg((p1.quantity+8)\*81,ios::beg);  while (getline(file, data1)) {  cout << data1 << endl;  }  }  else if(p1.quantity==4){  file.seekg((p1.quantity+7)\*69.5,ios::beg);  while (getline(file, data1)) {  cout << data1 << endl;  }  }  else if(p1.quantity==5){  file.seekg((p1.quantity+6)\*57.9,ios::beg);  while (getline(file, data1)) {  cout << data1 << endl;  }  }  else if(p1.quantity==6){  file.seekg((p1.quantity+5)\*46.3,ios::beg);  while (getline(file, data1)) {  cout << data1 << endl;  }  }  else if(p1.quantity==7){  file.seekg((p1.quantity+4)\*34.8,ios::beg);  while (getline(file, data1)) {  cout << data1 << endl;  }  }  else if(p1.quantity==8){  file.seekg((p1.quantity+3)\*23.2,ios::beg);  while (getline(file, data1)) {  cout << data1 << endl;  }  }  else if(p1.quantity==9){  file.seekg((p1.quantity+2)\*11.7,ios::beg);  while (getline(file, data1)) {  cout << data1 << endl;  }  }  else if(p1.quantity==10){  file.seekg((p1.quantity+1)\*0.2,ios::beg);  while (getline(file, data1)) {  cout << data1 << endl;  }  }  file.close();  } else {  cout << "We don't have that much products in our inventory(. ❛ ᴗ ❛.)" << endl;  }  } else {  cout << "we don't have that model ,you are trying to find (. ❛ ᴗ ❛.)" << endl;  }  }  }  fstream file1;  void buyingModelB(){  int quantity1=10;  product p1;  cout << "please tell us the model of the product you want to buy.." << endl;  cin >> p1.modelNum;  if (p1.modelNum == 431) {  if (quantity1 <= 0) {  cout << "Their is nothing in the inventory (ᵔᴥᵔ)" << endl;  exit(1);  }  cout << "How much quantity of products you want to buy from ten inventory products.." << endl;  cin >> p1.quantity;  if (p1.quantity <= quantity1) {  cout << "please enter product name." << endl;  cin >> p1.name;  file1.open("modelB.txt", ios::ate | ios::in);  string data2;  if (file1.is\_open()) {  cout << "==============you have bought the following things=============" << endl << endl;  if (p1.quantity == 1) {  file1.seekg((p1.quantity + 10) \* 104, ios::beg);  while (getline(file1, data2)) {  cout << data2 << endl;  }  } else if (p1.quantity == 2) {  file1.seekg((p1.quantity + 9) \* 92.5, ios::beg);  while (getline(file1, data2)) {  cout << data2 << endl;  }  } else if (p1.quantity == 3) {  file1.seekg((p1.quantity + 8) \* 81, ios::beg);  while (getline(file1, data2)) {  cout << data2 << endl;  }  } else if (p1.quantity == 4) {  file1.seekg((p1.quantity + 7) \* 69.5, ios::beg);  while (getline(file1, data2)) {  cout << data2 << endl;  }  } else if (p1.quantity == 5) {  file1.seekg((p1.quantity + 6) \* 57.9, ios::beg);  while (getline(file1, data2)) {  cout << data2 << endl;  }  } else if (p1.quantity == 6) {  file1.seekg((p1.quantity + 5) \* 46.3, ios::beg);  while (getline(file1, data2)) {  cout << data2 << endl;  }  } else if (p1.quantity == 7) {  file1.seekg((p1.quantity + 4) \* 34.8, ios::beg);  while (getline(file1, data2)) {  cout << data2 << endl;  }  } else if (p1.quantity == 8) {  file1.seekg((p1.quantity + 3) \* 23.2, ios::beg);  while (getline(file1, data2)) {  cout << data2 << endl;  }  } else if (p1.quantity == 9) {  file1.seekg((p1.quantity + 2) \* 11.7, ios::beg);  while (getline(file1, data2)) {  cout << data2 << endl;  }  } else if (p1.quantity == 10) {  file1.seekg((p1.quantity + 1) \* 0.2, ios::beg);  while (getline(file1, data2)) {  cout << data2 << endl;  }  }  file1.close();  }  else {  cout << "the file have nothing in it." << endl;  }  }  else {  cout << "We don't have that much products in our inventory(. ❛ ᴗ ❛.)" << endl;  }  }  else {  cout << "we don't have that model ,you are trying to find (. ❛ ᴗ ❛.)" << endl;  }  }  void buyingModelC(){  int quantity1=10;  product p1;  fstream file2;  cout << "please tell us the model of the product you want to buy.." << endl;  cin >> p1.modelNum;  if (p1.modelNum == 431) {  if (quantity1 <= 0) {  cout << "Their is nothing in the inventory (ᵔᴥᵔ)" << endl;  exit(1);  }  cout << "How much quantity of products you want to buy from 10 inventory products.." << endl;  cin >> p1.quantity;  if (p1.quantity <= quantity1) {  cout << "please enter product name." << endl;  cin >> p1.name;  file2.open("modelB.txt", ios::ate | ios::in);  string data3;  if (file2.is\_open()) {  cout<<"==============you have bought the following things============="<<endl<<endl;  if(p1.quantity==1){  file2.seekg((p1.quantity+10)\*104,ios::beg);  while (getline(file2, data3)) {  cout << data3 << endl;  }  }  else if(p1.quantity==2){  file1.seekg((p1.quantity+9)\*92.5,ios::beg);  while (getline(file2, data3)) {  cout << data3 << endl;  }  }  else if(p1.quantity==3){  file1.seekg((p1.quantity+8)\*81,ios::beg);  while (getline(file2, data3)) {  cout << data3 << endl;  }  }  else if(p1.quantity==4){  file1.seekg((p1.quantity+7)\*69.5,ios::beg);  while (getline(file2, data3)) {  cout << data3 << endl;  }  }  else if(p1.quantity==5){  file1.seekg((p1.quantity+6)\*57.9,ios::beg);  while (getline(file2, data3)) {  cout << data3 << endl;  }  }  else if(p1.quantity==6){  file1.seekg((p1.quantity+5)\*46.3,ios::beg);  while (getline(file2, data3)) {  cout << data3 << endl;  }  }  else if(p1.quantity==7){  file1.seekg((p1.quantity+4)\*34.8,ios::beg);  while (getline(file2, data3)) {  cout << data3 << endl;  }  }  else if(p1.quantity==8){  file1.seekg((p1.quantity+3)\*23.2,ios::beg);  while (getline(file2, data3)) {  cout << data3 << endl;  }  }  else if(p1.quantity==9){  file1.seekg((p1.quantity+2)\*11.7,ios::beg);  while (getline(file2, data3)) {  cout << data3 << endl;  }  }  else if(p1.quantity==10){  file1.seekg((p1.quantity+1)\*0.2,ios::beg);  while (getline(file2, data3)) {  cout << data3 << endl;  }  }  file2.close();  } else {  cout << "We don't have that much products in our inventory(. ❛ ᴗ ❛.)" << endl;  }  } else {  cout << "we don't have that model ,you are trying to find (. ❛ ᴗ ❛.)" << endl;  }  }  }  void CustomerBuyingMenu() {  int opt7;  do {  cout << "--------------choose options as a admin ----------------" << endl;  cout << "1.For buying product of modelA(431)." << endl;  cout << "2.For buying product of modelB(432)." << endl;  cout << "3.For buying product of modelC(433)." << endl;  cout << "4.exit." << endl;  cin >> opt7;  switch (opt7) {  case 1: {  buyingModelA();  break;  }  case 2: {  buyingModelB();  break;  }  case 3: {  buyingModelC();  break;  }  case 4: {  cout << "you choose to exit." << endl;  exit(1);  }  default: {  cout << "you have entered invalid option.";  }  }  } while (opt7 >= 1 && opt7 <= 4);  } |
| --- |

2. Write a method to replace all spaces in a string with '%2e: You may assume that the string has sufficient space at the end to hold the additional characters, and that you are given the "true" length of the string. (Note: In C++, please use a character array so that you can perform this operation in place.)

EXAMPLE

Input: "Mr John Smith JJ”, 13

Output: "Mr%2eJohn%2eSmith"

| int SpacesToCharacter(char arr[]) {  int countForSpaces = 0;  int i;  for (i = 0; arr[i]; i++) {  if (arr[i] == ' ') {  countForSpaces++;  }  }  while (arr[i - 1] == ' ') {  countForSpaces--;  i--;  }  int upgradedLength = ((i + (countForSpaces \* 2)) + 1);  int location = upgradedLength - 1;  if (upgradedLength > MAX) {  return -1;  }  arr[location--] = '\0';  for (int k = i - 1; k >= 0; k--) {  if (arr[k] == ' ') {  arr[location] = '0';  arr[location - 1] = '2';  arr[location - 2] = '%';  location = location - 3;  } else {  arr[location] = arr[k];  location--;  }  }  cout << "----> ";  return upgradedLength;  } |
| --- |

#### **Assessment Rubric for Lab**

**Method for assessment:**

Lab reports and instructor observation during lab sessions. Outcome assessed:

a. Ability to conduct experiments, as well as to analyze and interpret data (P) b. Ability to function on multi-disciplinary teams (A)

c. Ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (P)

| Performance metric | Mapping (task no. and description) | | Max marks | Exceeds expectation | Meets expectation | Does not meet expectation | Obtained marks |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Realization of experiment (a) | 1 | Functionality | 40 | Executes without errors excellent user prompts, good use of symbols, spacing in output. Through testing has been completed (35-40) | Executes without errors, user prompts are understandable, minimum use of symbols or spacing in output. Some testing has been completed (20-34) | Does not execute due to syntax errors, runtime errors, user prompts are misleading or non-existent. No testing has been completed (0-19) |  |
| 2. Teamwork (b) | 1 | Group Performance | 5 | Actively engages and cooperates with other group member(s) in effective manner (4-5) | Cooperates with other group member(s) in a reasonable manner but conduct can be improved (2-3) | Distracts or discourages other group members from conducting the experiment (0-1) |  |
| 3. Conducting experiment (a, c) | 1 | On Spot Changes | 10 | Able to make changes (8-10) | Partially able to make changes (5-7) | Unable to make changes (0-4) |  |
| 2 | Viva | 10 | Answered all questions (8-10) | Few incorrect answers (5-7) | Unable to answer all questions (0-4) |  |
| 4. Laboratory safety and disciplinary rules (a) | 1 | Code commenting | 5 | Observes lab safety rules; handles the equipment and parts with care and adheres to the lab disciplinary guidelines aptly (4-5) | Generally observes safety rules and disciplinary guidelines with minor lapses (2-3) | Disregards lab safety and disciplinary rules (0-1) |  |
| 5. Data collection (c) | 1 | Code Structure | 5 | Excellent use of white space, creatively organized work, excellent use of variables and constants, correct identifiers for constants, No line-wrap (4-5) | Includes name, and assignment, white space makes the program fairly easy to read. Title, organized work, good use of variables (2-3) | Poor use of white space (indentation, blank lines) making code hard to read, disorganized and messy (0-1) |  |
| 6. Data analysis (a, c) | 1 | Algorithm | 20 | Solution is efficient, easy to understand, and maintain (15-20) | A logical solution that is easy to follow but it is not the most efficient (6-14) | A difficult and inefficient solution (0-5) |  |
| 7. Computer use (c) | 1 | Documentation | 5 | Timely documented (4-5) | Late documented (2-3) | Not documented (0-1) |  |
|  | Max Marks (total): | | 100 | Obtained Marks (total): | | |  |

Lab Engineer Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_