Q1:

bookTypeMain.cpp

#include <iostream>  
#include "bookTypeImp.h"  
  
using namespace std;  
  
int main()  
{  
//Prompt the message on a screen  
 cout << "\n\*\*\*\*\*\*\*\*\*\*\*Books as an ADT bookType\*\*\*\*\*\*\*\*\*\*\*\*\*";  
  
//Declare an array of 100  
//components of type bookType  
 bookType numBooks[100];  
  
//Declare variables  
 string str;  
 double price;  
 int copies;  
 char userChoice;  
 int count = 0;  
  
 do  
 {  
 //Prompt and read the input from the user  
 cout << "\n\n\tEnter the book Title: ";  
 cin >> str;  
 numBooks[count].setTitle(str);  
 int j = 0;  
 do  
 {  
 cout << "\tEnter the author Name : ";  
 cin >> str;  
 numBooks[count].setAuthor(str);  
 j++;  
 cout << "\n\tMore authors ( Y/N ) ? ";  
 cin >> userChoice;  
 } while (j < 4 && tolower(userChoice) != 'n');  
 cout << "\tEnter publisher name: ";  
 cin >> str;  
 numBooks[count].setPublisher(str);  
 cout << "\tEnter the ISBN Number: ";  
 cin >> str;  
 numBooks[count].setISBN(str);  
 cout << "\n\tEnter the price: ";  
 cin >> price;  
 numBooks[count].setPrice(price);  
  
 // Prompt and read number of copies  
 cout << "\tEnter copies : ";  
 cin >> copies;  
 numBooks[count].setCopies(copies);  
 count++;  
 cout << "\n\tEnter more books (Y/N) ? ";  
 cin >> userChoice;  
 // Convert to lower case  
 if (userChoice < 92)  
 userChoice += 32;  
  
 } while (userChoice != 'n'); // end do-while  
  
 do  
 {  
 // search for a book using title and ISBN  
 cout << "\n\tSearch for a book by its title: T";  
 cout << "\n\tSearch for a book by its ISBN : I";  
 cout << "\n\tUpdate the number of copies : U";  
 cout << "\n\tExit : E";  
 cout << "\n\n\tEnter your choice : ";  
 cin >> userChoice;  
 int i;  
 char T, I, U, E;  
 switch (toupper(userChoice))  
 {  
 // Search for a book by its title  
 case 'T': cout << "\n\tEnter book title to search : ";  
 cin >> str;  
  
 for (i = 0; i < count; i++)  
 if (numBooks[i].compareTitle(str))  
 break;  
  
 if (i == count)  
 cout << "\n\tBook not found.";  
 else  
 cout << "\n\tBook found.";  
 break;  
  
 // search for a book by its ISBN number  
 case 'I':cout << "\n\tEnter ISBN number to search : ";  
 cin >> str;  
  
 for (i = 0; i < count; i++)  
 if (numBooks[i].compareISBN(str))  
 break;  
  
 if (i == count)  
 cout << "\n\tBook not found.";  
 else  
 cout << "\n\tBook found.";  
  
 break;  
  
 // Update the copies of the book  
 case 'U': cout << "\n\tEnter book title to search : ";  
 cin >> str;  
  
 for (i = 0; i < count; i++)  
 if (numBooks[i].compareTitle(str))  
 break;  
  
 if (i == count)  
 cout << "\n\tBook not found.";  
 else  
 {  
 cout << "\n\tEnter the number of copies ";  
 cin >> copies;  
 numBooks[i].updateCopies(copies);  
 } break;  
 //For exit the choice  
 case 'E':  
 return 0;  
 }  
 } while (true);  
 return 0;  
}

bookTypeImp.h

#ifndef Q1\_BOOKTYPEIMP\_H  
#define Q1\_BOOKTYPEIMP\_H  
  
#include<string>  
  
using namespace std;  
  
class bookType  
{  
public:  
 bookType();  
 void setTitle(string);  
 string getTile();  
 bool compareTitle(string);  
  
 void setAuthor(string = "");  
 void showAuthors();  
 void updateAuthor(string = "");  
 string \*getAuthors();  
  
 void setCopies(int);  
 void showCopies();  
 void updateCopies(int);  
 int getCopies();  
  
  
 void setPublisher(string);  
 void showPublisher();  
 void updatePublisher(string);  
 string getPublisher();  
  
 void setISBN(string);  
 void showISBN();  
 void updateISBN(string);  
 string getISBN();  
 bool compareISBN(string);  
  
 void setPrice(double);  
 void showPrice();  
 void updatePrice(double);  
 double getPrice();  
  
private:  
 string title;  
 string authors[4];  
 string publisher;  
 string ISBN;  
 double price;  
 int copies;  
 int authorsNo;  
};  
  
  
#endif //Q1\_BOOKTYPEIMP\_H

bookTypeImp.cpp

#include "bookTypeImp.h"  
#include<iostream>  
  
using namespace std;  
  
bookType::bookType()  
{  
 title = "";  
 for (int i = 0; i < 4; i++)  
 authors[i] = "";  
 publisher = "";  
 ISBN = "";  
 price = 0;  
 copies = 0;  
 authorsNo = 0;  
}  
//setter and getter methods  
//Method definition of setTitle  
void bookType::setTitle(string myTitle)  
{  
 title = myTitle;  
}  
//Method definition of getTitle  
string bookType::getTile()  
{  
 return title;  
}  
//Method definition of compareTitle  
bool bookType::compareTitle(string otherTitle)  
{  
 return (title.compare(otherTitle) == 0);  
}  
//Method definition of setAuthor  
void bookType::setAuthor(string myAuthor)  
{  
 authorsNo = authorsNo % 4;  
 if (myAuthor.compare("") == 0)  
 return;  
 else  
 {  
 authors[authorsNo] = myAuthor;  
  
 authorsNo++;  
 }  
}  
//Method definition of showAuthors  
void bookType::showAuthors()  
{  
 for (int i = 0; i < authorsNo; i++)  
 cout << authors[i] << ", ";  
 cout << "\r\r";  
}  
//Method definition of updateAuthor  
void bookType::updateAuthor(string myAuthor)  
{  
 setAuthor(myAuthor);  
}  
//Method definition of getAuthors  
string \*bookType::getAuthors()  
{  
 return authors;  
}  
//Method definition of setCopies  
void bookType::setCopies(int myCopies)  
{  
 copies = myCopies;  
}  
//Method definition of showCopies  
void bookType::showCopies()  
{  
 cout << "\n\tThe number of copies " << copies;  
}  
//Method definition of updateCopies  
void bookType::updateCopies(int myCopies)  
{  
 copies = myCopies;  
}  
//Method definition of getCopies  
int bookType::getCopies()  
{  
 return copies;  
}  
//Method definition of setPublisher  
void bookType::setPublisher(string myPublisher)  
{  
 publisher = myPublisher;  
}  
//Method definition of showPublisher  
void bookType::showPublisher()  
{  
 cout << publisher;  
}  
//Method definition of updatePublisher  
void bookType::updatePublisher(string myPublisher)  
{  
 publisher = myPublisher;  
}  
//Method definition of getPublisher  
string bookType::getPublisher()  
{  
 return publisher;  
} // end function getPublisher  
  
void bookType::setISBN(string myISBN)  
{  
 ISBN = myISBN;  
}  
//Method definition of showISBN  
void bookType::showISBN()  
{  
 cout << ISBN;  
}  
//Method definition of updateISBN  
void bookType::updateISBN(string myISBN)  
{  
 ISBN = myISBN;  
}  
//Method definition of getISBN  
string bookType::getISBN()  
{  
 return ISBN;  
}  
//Method definition of compareISBN  
bool bookType::compareISBN(string myISBN)  
{  
 return (myISBN.compare(ISBN) == 0);  
}  
//Method definition of setPrice  
void bookType::setPrice(double myPrice)  
{  
 price = myPrice;  
}  
//Method definition of showPrice  
void bookType::showPrice()  
{  
 cout << "\n\tThe book price is " << price;  
}  
//Method definition of updatePrice  
void bookType::updatePrice(double myPrice)  
{  
 price = myPrice;  
}//Method definition of getPrice  
double bookType::getPrice()  
{  
 return price;  
}

Enter the book Title: C++

Enter the author Name : Felicity

More authors ( Y/N ) ? Y

Enter the author Name : Lily

More authors ( Y/N ) ? N

Enter publisher name: Wiley

Enter the ISBN Number: 123456789

Enter the price: 800

Enter copies : 120

Enter more books (Y/N) ? Y

Enter the book Title: Java

Enter the author Name : Vivian

More authors ( Y/N ) ? N

Enter publisher name: Westcost

Enter the ISBN Number: 987654321

Enter the price: 500

Enter copies : 120

Enter more books (Y/N) ? N

Search for a book by its title: T

Search for a book by its ISBN : I

Update the number of copies : U

Exit : E

Enter your choice : T

Enter book title to search : C++

Book found.

Search for a book by its title: T

Search for a book by its ISBN : I

Update the number of copies : U

Exit : E

Enter your choice : I

Enter ISBN number to search : 123456789

Book found.

Search for a book by its title: T

Search for a book by its ISBN : I

Update the number of copies : U

Exit : E

Enter your choice : U

Enter book title to search : Java

Enter the number of copies 600

Search for a book by its title: T

Search for a book by its ISBN : I

Update the number of copies : U

Exit : E

Enter your choice : T

Enter book title to search : Python

Book not found.

Search for a book by its title: T

Search for a book by its ISBN : I

Update the number of copies : U

Exit : E

Enter your choice : E

Process finished with exit code 0

Q2:

bookType.cpp

#include "bookType.h"  
#include <iostream>  
#include <string>  
using namespace std;  
  
//default constructor that sets each data member blank  
bookType::bookType() {  
 title = "";  
 author1 = "";  
 author2 = "";  
 author3 = "";  
 author4 = "";  
 publisher = "";  
 isbn = "";  
 price = 0.0;  
 numCopiesInStock = 0;  
 numAuthors = 0;  
}  
  
//default constructor that sets each data member according to the parameters  
bookType::bookType(string t, string a1, string a2, string a3, string a4,  
 string p, string i, double pr, int nc, int na) {  
  
 title = t;  
 author1 = a1;  
 author2 = a2;  
 author3 = a3;  
 author4 = a4;  
 publisher = p;  
 isbn = i;  
 price = pr;  
 numCopiesInStock = nc;  
 numAuthors = na;  
}  
  
//prints the title of the book  
void bookType::showTitle() {  
  
 cout << "Title: " << title << endl;  
}  
  
//sets the title of the book according to the parameter  
void bookType::setTitle(string otherTitle) {  
  
 title = otherTitle;  
}  
  
//checks to see if another title is equal to the current title  
bool bookType::checkTitle(string otherTitle) {  
  
 if (title == otherTitle) {  
 return true;  
 }  
 else {  
 return false;  
 }  
}  
  
//prints the number of copies in stock  
void bookType::showNumCopiesInStock() {  
  
 cout << "Number of copies in stock: " << numCopiesInStock << endl;  
}  
  
//sets the number of copies in stock according to the parameter  
void bookType::setNumCopiesInStock(int num) {  
  
 numCopiesInStock = num;  
}  
  
//returns the number of copies in stock  
int bookType::returnNumCopiesInStock() {  
  
 return numCopiesInStock;  
}  
  
//prints the publisher name  
void bookType::showPublisher() {  
  
 cout << "Publisher: " << publisher << endl;  
}  
  
//sets the publisher name according to the parameter  
void bookType::setPublisher(string name) {  
  
 publisher = name;  
}  
  
//returns the publisher name  
string bookType::returnPublisher() {  
  
 return publisher;  
}  
  
//prints the isbn string  
void bookType::showIsbn() {  
  
 cout << "ISBN: " << isbn << endl;  
}  
  
//sets the isbn string according to the parameter  
void bookType::setIsbn(string num) {  
  
 isbn = num;  
}  
  
//returns the isbn  
string bookType::returnIsbn() {  
  
 return isbn;  
}  
  
//checks to see if another isbn is equal to the current isbn  
bool bookType::checkIsbn(string num) {  
  
 if (isbn == num) {  
 return true;  
 }  
 else {  
 return false;  
 }  
}  
  
//prints the book price  
void bookType::showBookPrice() {  
  
 cout << "Price: $" << price << endl;  
}  
  
//sets the book price according to the parameter  
void bookType::setBookPrice(double num) {  
  
 price = num;  
}  
  
//returns the book price  
double bookType::returnBookPrice() {  
  
 return price;  
}  
  
//prints the authors  
void bookType::showAuthors() {  
  
 cout << "Authors: " << author1 << " " << author2 << " " << author3 << " " << author4 << endl;  
}  
  
//sets the author, with the author# and a name  
void bookType::setAuthor(int a, string name) {  
  
 switch (a){  
 case 1:  
 author1 = name;  
 break;  
 case 2:  
 author2 = name;  
 break;  
 case 3:  
 author3 = name;  
 break;  
 case 4:  
 author4 = name;  
 break;  
 default:  
 cout << "Invalid entry. Setting the first author." << endl;  
 author1 = name;  
 }  
}  
  
//returns the author based on the number given  
string bookType::returnAuthor(int a) {  
  
 switch (a) {  
 case 1:  
 return author1;  
 break;  
 case 2:  
 return author2;  
 break;  
 case 3:  
 return author3;  
 break;  
 case 4:  
 return author4;  
 break;  
 default:  
 cout << "Invalid entry. Returning the first author." << endl;  
 return author1;  
 }  
}  
  
//prints all of the details of the bookType  
void bookType::print() {  
  
 showTitle();  
 showAuthors();  
 showPublisher();  
 showIsbn();  
 showBookPrice();  
 showNumCopiesInStock();  
}

bookType.h

#ifndef Q2\_BOOKTYPE\_H  
#define Q2\_BOOKTYPE\_H  
#pragma once  
#include <iostream>  
#include <string>  
using namespace std;  
  
class bookType {  
  
public:  
  
 bookType(); //default constructor with no parameters  
 bookType(string t, string a1, string a2, string a3, string a4, string p,  
 string i, double pr, int nc, int na); //default constructor with all parameters  
  
 void print(); //prints the data of a bookType  
  
 void showTitle(); //prints the title  
 void setTitle(string otherTitle); //sets the title  
 bool checkTitle(string otherTitle); //checks to see if another title is equal to the current title  
  
 void showNumCopiesInStock(); //prints the number of copies in stock  
 void setNumCopiesInStock(int num); //sets the number of copies in stock  
 int returnNumCopiesInStock(); //returns the number of copies in stock  
  
 void showPublisher(); //prints the title  
 void setPublisher(string name); //sets the title  
 string returnPublisher(); //returns the title  
  
 void showIsbn(); //prints the isbn  
 void setIsbn(string num); //sets the isbn  
 string returnIsbn(); //returns the isbn  
 bool checkIsbn(string num); //checks to see if another isbn is equal to the current isbn  
  
 void showBookPrice(); //prints the book price  
 void setBookPrice(double num); //sets the book price  
 double returnBookPrice(); //returns the book price  
  
 void showAuthors(); //prints the authors  
 void setAuthor(int a, string name); //sets the author with a number for the author#, and a name  
 string returnAuthor(int a); //returns the author by their author#  
  
private:  
  
 string title, //holds the title name  
 author1, //holds the name of author1  
 author2, //holds the name of author2  
 author3, //holds the name of author3  
 author4, //holds the name of author4  
 publisher,//holds the publisher name  
 isbn; //holds the isbn string  
  
 double price; //holds the price of the book  
  
 int numCopiesInStock, //holds the number of copies in stock  
 numAuthors; //holds the number of authors  
};  
  
  
#endif //Q2\_BOOKTYPE\_H

Main.cpp

#include "bookType.h"  
#include <iostream>  
#include <string>  
#include <vector>  
using namespace std;  
  
int findByTitle(vector<bookType> list, string t); //function to search a vector for a book title  
int findByIsbn(vector<bookType> list, string i); //function to search a vector for a book isbn  
  
int main() {  
  
 int position; //holds the index when searching for isbn or title  
 vector<bookType> bookList; //vector to hold elements of type bookType  
 bookType book1, book2, book3; //3 bookTypes that will be initialized  
  
 //initializing book 1:  
 book1.setTitle("The Hunger Games");  
 book1.setAuthor(1, "Suzanne Collins");  
 book1.setPublisher("Scholastic Press");  
 book1.setIsbn("9780439023481");  
 book1.setBookPrice(25.99);  
 book1.setNumCopiesInStock(15);  
  
 //initializing book2:  
 book2.setTitle("To Kill A Mockingbird");  
 book2.setAuthor(1, "Harper Lee");  
 book2.setPublisher("J. B. Lippincott & Co.");  
 book2.setIsbn("9780446310789");  
 book2.setBookPrice(15.99);  
 book2.setNumCopiesInStock(7);  
  
 //initializing book3:  
 book3.setTitle("Catch-22");  
 book3.setAuthor(1, "Joseph Heller");  
 book3.setPublisher("Simon & Schuster");  
 book3.setIsbn("9780099529118");  
 book3.setBookPrice(18.99);  
 book3.setNumCopiesInStock(40);  
  
 //adding the books to the vector:  
 bookList.push\_back(book1); //First Method  
 bookList.push\_back(book2);  
  
 // adding book3 after book1  
 vector<bookType>::iterator it = bookList.begin() + 1;  
 bookList.insert( it, book3); // Second method  
  
  
 //printing each bookType in the vector:  
 cout << "Printing All Of The Books In The Book List..." << endl;  
 for (int i = 0; i < bookList.size(); i++) { //steps through each bookType in the bookList  
 cout << endl;  
 bookList[i].print(); //prints the bookType and its data  
 }  
  
 cout << endl;  
 cout << "Searching For Title Catch-22..." << endl;  
  
 //searching the bookList for the title Catch-22:  
 position = findByTitle(bookList, "Catch-22"); //sets position  
 if (position == -1) { //if title not in the list...  
 cout << "Catch-22 is not in the book list." << endl;  
 }  
 else { //else, the title is in the list...  
 cout << "Catch-22 is in the book list, and at position " << position << "." << endl;  
 }  
  
 cout << endl;  
 cout << "Searching For ISBN 19020483547682..." << endl;  
  
 //searching the bookList for the isbn 19020483547682...  
 position = findByIsbn(bookList, "19020483547682"); //sets position  
 if (position == -1) { //if isbn not in the list  
 cout << "19020483547682 is not in the book list." << endl;  
 }  
 else { //else, the isbn is in the list  
 cout << "19020483547682 is in the book list, and at position " << position << "." << endl;  
 }  
  
 cout << "The size of the book list is: " << bookList.size() << endl; // third Method  
  
 cout << "-----------" << endl;  
  
 if (bookList.empty()) // Fourth Method  
 cout << "The book list is empty\n";  
 else  
 cout << "The book list is not empty\n";  
 bookList.pop\_back(); // Fifth Method  
 cout << "Printing All Of The Books In The Book List now..." << endl;  
 for (int i = 0; i < bookList.size(); i++) { //steps through each bookType in the bookList  
 cout << endl;  
 bookList[i].print(); //prints the bookType and its data  
 }  
 return 0;  
}  
  
//returns the index of a bookType in a vector, else returns -1  
int findByTitle(vector<bookType> list, string t) {  
  
 int position = -1;  
 for (int i = 0; i < list.size(); i++) { //steps through the vector  
  
 if (list[i].checkTitle(t)) { //checks each bookType to see if the title matches  
 position = i;  
 }  
 }  
 return position; //returns the index  
}  
  
//returns the isbn of a bookType in a vector, else returns -1  
int findByIsbn(vector<bookType> list, string is) {  
  
 int position = -1;  
 for (int i = 0; i < list.size(); i++) { //steps through the vector  
  
 if (list[i].checkIsbn(is)) { //checks each bookType to see if the isbn matches  
 position = i;  
 }  
 }  
 return position; //returns the index  
}

Printing All Of The Books In The Book List...

Title: The Hunger Games

Authors: Suzanne Collins

Publisher: Scholastic Press

ISBN: 9780439023481

Price: $25.99

Number of copies in stock: 15

Title: Catch-22

Authors: Joseph Heller

Publisher: Simon & Schuster

ISBN: 9780099529118

Price: $18.99

Number of copies in stock: 40

Title: To Kill A Mockingbird

Authors: Harper Lee

Publisher: J. B. Lippincott & Co.

ISBN: 9780446310789

Price: $15.99

Number of copies in stock: 7

Searching For Title Catch-22...

Catch-22 is in the book list, and at position 1.

Searching For ISBN 19020483547682...

19020483547682 is not in the book list.

The size of the book list is: 3

-----------

The book list is not empty

Printing All Of The Books In The Book List now...

Title: The Hunger Games

Authors: Suzanne Collins

Publisher: Scholastic Press

ISBN: 9780439023481

Price: $25.99

Number of copies in stock: 15

Title: Catch-22

Authors: Joseph Heller

Publisher: Simon & Schuster

ISBN: 9780099529118

Price: $18.99

Number of copies in stock: 40

Process finished with exit code 0

https://www.cplusplus.com/reference/vector/vector/