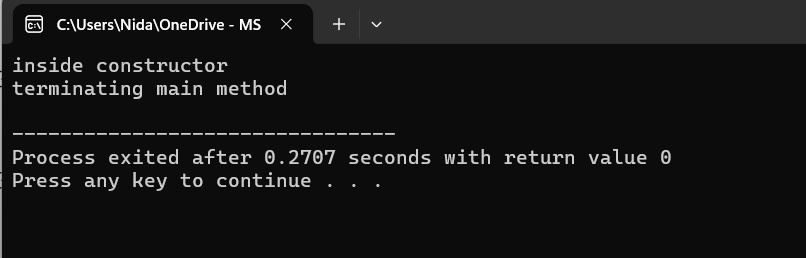
**Question 1**

Part 1



Part 2:

Line 6 [Error] default argument missing for parameter 2 of 'ABC::ABC(int, int)'

Line 10 [Error] missing terminating " character

Part 3: any valid answer in one line

Part 4: any valid answer in one line

Q2 (a)

#include <iostream>

using namespace std;

class Book {

int bookId;

string title, author;

public:

Book() : bookId(0), title(""), author("") {}

Book(int id, string t, string a) : bookId(id), title(t), author(a) {}

int getId() { return bookId; }

void display() {

cout << "Book ID: " << bookId << ", Title: " << title << ", Author: " << author << endl;

}

};

class Library {

int libraryId, bookCount, capacity;

string libraryName;

Book\* books;

public:

Library(int id, string name, int size) : libraryId(id), libraryName(name), bookCount(0), capacity(size) {

books = new Book[capacity];

}

void addBook(int id, string title, string author) {

if (bookCount < capacity) {

books[bookCount] = Book(id, title, author);

bookCount++;

} else {

cout << "Library is full! Cannot add more books." << endl;

}

}

void removeBook(int id) {

bool found = false;

for (int i = 0; i < bookCount; i++) {

if (books[i].getId() == id) {

found = true;

for (int j = i; j < bookCount - 1; j++) {

books[j] = books[j + 1];

}

bookCount--;

cout << "Book with ID " << id << " removed." << endl;

break;

}

}

if (!found) {

cout << "Book not found!" << endl;

}

}

void displayLibrary() {

cout << "Library ID: " << libraryId << ", Name: " << libraryName << endl;

for (int i = 0; i < bookCount; i++) {

books[i].display();

}

}

~Library() {

delete[] books;

cout << "Library destructed and memory freed." << endl;

}

};

int main() {

Library lib(1, "City Library", 3);

lib.addBook(101, "The Great Gatsby", "F. Scott Fitzgerald");

lib.addBook(102, "To Kill a Mockingbird", "Harper Lee");

lib.displayLibrary();

lib.removeBook(101);

lib.displayLibrary();

return 0;

}

**Q2 (b) composition**