# EXPERIMENT NUMBER –final practical exam worksheet

Questions -

a) WAP to allocate memory dynamically for an object of a given class using class’s constructor.

b) WAP to maintain book record file handling.

**Program code**

#include <bits/stdc++.h>

#include <conio.h>

#include <fstream>

using namespace std;

// Book class

class Book {

private:

int bookid;

// Max book title size 20

char title[20];

float price;

public:

// Default Constructor

Book()

{

bookid = 0;

strcpy(title, "no title");

price = 0;

}

// Function for taking book data

void getBookData()

{

cout << "Enter bookid, title, "

<< " price: ";

cin >> bookid;

cin.ignore();

cin.getline(title, 19);

cin >> price;

}

// Function for showing book data

void showBookData()

{

cout << "\n"

<< bookid

<< " " << title << " " << price;

}

int storeBook();

void viewAllBooks();

void searchBook(char\*);

void deleteBook(char\*);

void updateBook(char\*);

};

// Function for update book data

void Book::updateBook(char\* t)

{

fstream file;

// Open the file

file.open("myfile3.txt",

ios::in | ios::out | ios::ate | ios::binary);

file.seekg(0);

file.read((char\*)this, sizeof(\*this));

// Read the file

while (!file.eof()) {

if (!strcmp(t, title)) {

getBookData();

file.seekp(file.tellp() - sizeof(\*this));

file.write((char\*)this, sizeof(\*this));

}

file.read((char\*)this, sizeof(\*this));

}

// Close the file

file.close();

}

// Function for delete book data

void Book::deleteBook(char\* t)

{

ifstream fin;

ofstream fout;

fin.open("myfile3.txt",

ios::app | ios::binary);

if (!fin)

cout << "\n File not found";

else {

fout.open("tempfile.txt",

ios::app | ios::binary);

fin.read((char\*)this, sizeof(\*this));

// Until end of file is not reached

while (!fin.eof()) {

if (strcmp(title, t))

fout.write((char\*)this, sizeof(\*this));

fin.read((char\*)this, sizeof(\*this));

}

fin.close();

fout.close();

remove("myfile3.txt");

rename("tempfile.txt", "myfile3.txt");

}

}

// Function to search the Book

void Book::searchBook(char\* t)

{

int counter = 0;

ifstream fin;

fin.open("myfile3.txt",

ios::in | ios::binary);

// If file is not found

if (!fin)

cout << "File not found";

else {

fin.read((char\*)this, sizeof(\*this));

// Until end of file is not reached

while (!fin.eof()) {

if (!strcmp(t, title)) {

showBookData();

counter++;

}

fin.read((char\*)this, sizeof(\*this));

}

if (counter == 0)

cout << "No record found";

fin.close();

}

}

// Function to view all the Book

void Book::viewAllBooks()

{

ifstream fin;

// Open function open file named

// myfile.txt

fin.open("myfile3.txt", ios::in | ios::binary);

if (!fin)

cout << "File not found";

else {

fin.read((char\*)this, sizeof(\*this));

// Until end of file is

// not reached

while (!fin.eof()) {

showBookData();

// read is object of ifstream

// class which is used for

// read data of file

fin.read((char\*)this, sizeof(\*this));

}

fin.close();

}

}

// Function to implement the store of

// all the books

int Book::storeBook()

{

if (bookid == 0 && price == 0) {

cout << "book data not"

<< " initialized";

return (0);

}

else {

ofstream fout;

// In the below line open function

// used to open files. If the file

// does not exist then it will

// create the file "myfile.txt"

fout.open("myfile3.txt", ios::app | ios::binary);

// Write function is used for

// data to write in the file

fout.write((char\*)this, sizeof(\*this));

fout.close();

return (1);

}

}

// Function to display the menu of the

// current menu-driver

int menu()

{

int choice;

cout<<"==| smita shinde uid- 20BCS4643 |=="<<endl;

cout << "\nBook Management";

cout << "\n1.Insert book record";

cout << "\n2.View all book record";

cout << "\n3.Search book record";

cout << "\n4.Delete book record";

cout << "\n5.Update book record";

cout << "\n6.Exit";

cout << "\n\nEnter your choice : ";

cin >> (choice);

return (choice);

}

// Driver Code

int main()

{

// Object of the class Book

Book b1;

char title[20];

while (1) {

system("cls");

switch (menu()) {

case 1:

b1.getBookData();

b1.storeBook();

cout << "\nRecord inserted";

break;

case 2:

b1.viewAllBooks();

break;

case 3:

cout << "\nEnter title of "

<< "book to search : ";

cin.ignore();

cin.getline(title, 19);

b1.searchBook(title);

break;

case 4:

cout << "\nEnter book title "

<< "for delete record : ";

cin.ignore();

cin.getline(title, 19);

b1.deleteBook(title);

break;

case 5:

cout << "\nEnter book title "

<< "to update record : ";

cin.ignore();

cin.getline(title, 19);

b1.updateBook(title);

break;

case 6:

cout << "\n Thanks for using";

cout << "\n Press any key to exit";

getch();

exit(0);

default:

cout << "Invalid input";

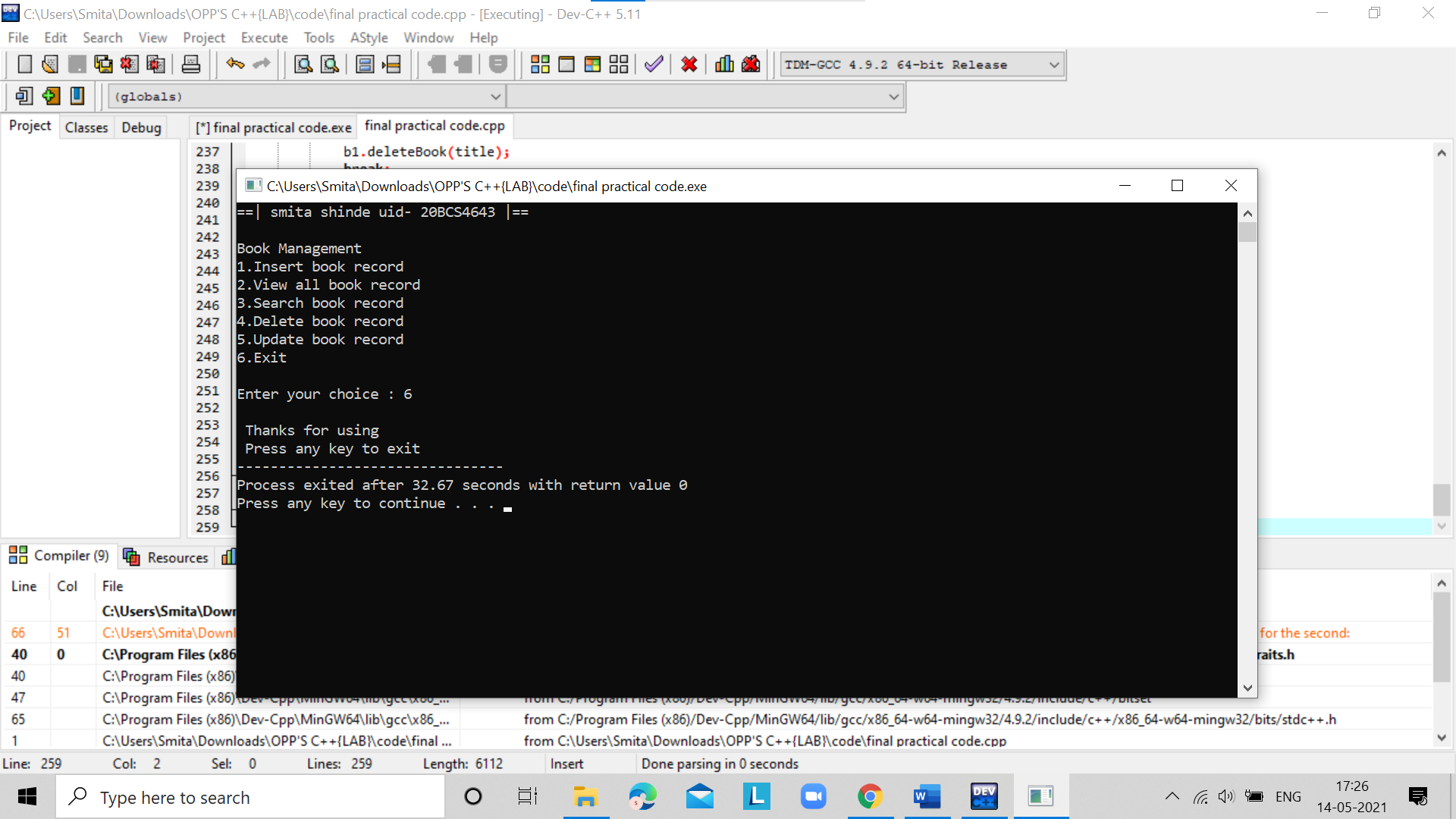
}

getch();

}

}

**OUTPUT**



# Question no.2nd

**TOPIC OF EXPERIMENT** – WAP to allocate memory dynamically for an object of a given class using class’s constructor.

**FLOWCHART/ ALGORITHM**

Start.

Step 1→ Creating a header file for input output stream and define the context.

# Step 2 → After that used using namespace std;

Step 3 → Create the class name followed by class student.

Step 4 → Creating the one constructor name followed by class name and one destructor name followed by class name with (~) operator.

# Step 5 → Declare a destructor ~student() for deallocating the memory allocated to the constructor.

Step 6 →Creating int main() dynamic memory will be allocated using the new statement.

Stop.

**PROGRAM CODE**

#include <iostream>

using namespace std;

class student

{

public:

student()

{

cout<< "Constructor Used" <<endl;

}

~student()

{

cout<< "Destructor Used" <<endl;

}

};

int main()

{

cout<<"==|smita shinde uid- 20bcs4643|=="<<endl;

student\* S = new student[3];

delete[]S;

}

**ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION**

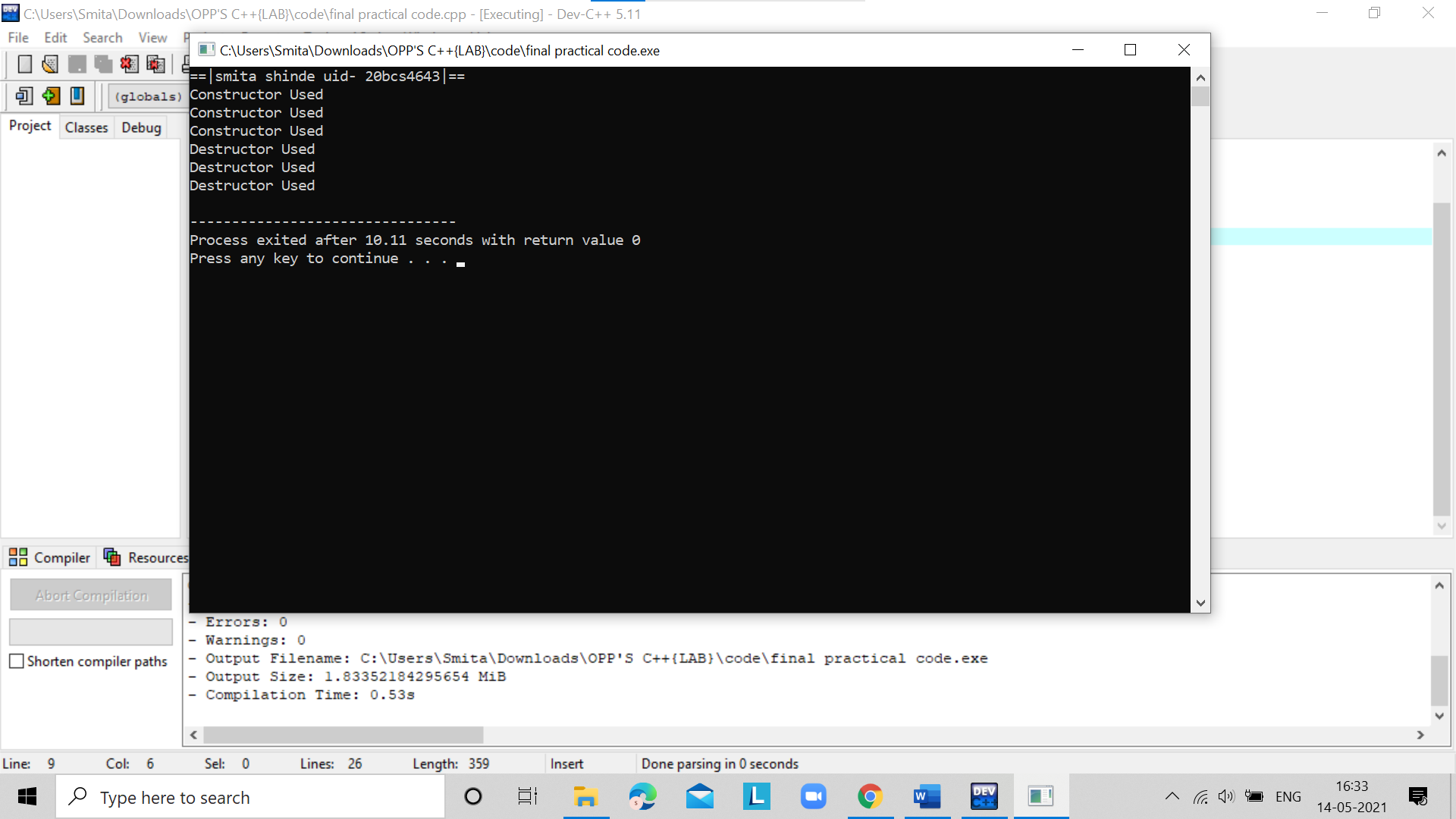
**(Kindly jot down the compile time errors encountered)**

No errors encountered

**PROGRAMS’ EXPLANATION (in brief)**

1. We start our program with preprocessor (#) and header file (< iostream >) we have many types of header files but in this program, we used iostream. #include< iostream >, Int main () is a function which work as a container of statements. All the statements are enclosed within the pair of braces { }. “using namespace std” means we use the namespace named std. “std” is an abbreviation for standard. So that means we use all the things with in “std” namespace.
2. we create a class student in which we create two member functions on is constructor and another is destructor. Then in the main function we create a pointer object and initialized it to array of numbers with new keyword. New keyword is used for Dynamic memory allocation During run time memory is allocated to each object and with the allocation of memory objects will created and then our constructor and destructor called automatically. Then we use delete keyword for cleaning the memory are the execution of the object and in the end, we execute our program.

**OUTPUT**



**LEARNING OUTCOMES**

| * Identify situations where computational methods would be useful. |
| --- |
| * Approach the programming tasks using techniques learnt and write pseudo-code. |
| * Choose the right data representation formats based on the requirements of the problem. |
| * Use the comparisons and limitations of the various programming constructs and choose the right one for the task. |

**EVALUATION COLUMN (To be filled by concerned faculty only)**

| **Sr. No.** | **Parameters** | **Maximum**  **Marks** | **Marks**  **Obtained** |
| --- | --- | --- | --- |
| 1. | Worksheet Completion including writing learning objective/ Outcome | 10 |  |
| 2. | Post Lab Quiz Result | 5 |  |
| 3. | Student engagement in Simulation/ Performance/ Pre Lab Questions | 5 |  |
| 4. | Total Marks | 20 |  |

In the above program, the coefficients a, b, and c are set to 2.3, 4, and 5.6 respectively. Then, the determinant is calculated as b2 - 4ac.

Based on the value of the determinant, the roots are calculated as given in the formula above. Notice we've used library function Math.sqrt() to calculate the square root of a number.

We have used the format() method to print the calculated roots.

The format() function can also be replaced by printf() as: