**task 2:**

// Class.h file (class header file)

#pragma once

#ifndef CLASS2\_H

#include<string>

#include<iostream>

using namespace std;

#define CLASS2\_H

#endif

class MultimediaItem

{

private:

string title;

string creator;

int publicationYear;

public:

MultimediaItem(string t, string c, int py) {

title = t;

creator = c;

publicationYear = py;

}

void setTitle(string t);

string getTitle();

void setCreator(string c);

string getCreator();

void setPublicationYear(int py);

int getPublicationYear();

};

class Book : public MultimediaItem

{

private:

string author;

string genre;

int number\_of\_pages;

public:

Book(string t, string c, int py, string a, string g, int np) : MultimediaItem(t, c, py) {

author = a;

genre = g;

number\_of\_pages = np;

}

void setAuthor(string a);

string getAuthor();

void setGenre(string g);

string getGenre();

void setnumber\_of\_pages(int np);

int getnumber\_of\_pages();

friend void displayDetails(Book b);

};

class Video : public MultimediaItem

{

private:

string format;

int duration;

int resolution;

public:

Video(string t, string c, int py, string f, int d, int r) : MultimediaItem(t, c, py) {

format = f;

duration = d;

resolution = r;

}

void setFormat(string f);

string getFormat();

void setDuration(int d);

int getDuration();

void setResolution(int r);

int getResolution();

friend void displayDetails(Video v);

};

class MusicAlbum : public MultimediaItem

{

private:

string artist;

string genre;

int numTracks;

public:

MusicAlbum(string t, string c, int py, string ar, string g, int nt) : MultimediaItem(t, c, py) {

artist = ar;

genre = g;

numTracks = nt;

}

void setArtist(string ar);

string getArtist();

void setGenre(string g);

string getGenre();

void setNumTracks(int nt);

int getNumTracks();

friend void displayDetails(MusicAlbum m);

};

//Class.cpp file (class source file)

#include "interface2.h"

void MultimediaItem::setTitle(string t)

{

title = t;

}

string MultimediaItem::getTitle()

{

return title;

}

void MultimediaItem::setCreator(string c)

{

creator = c;

}

string MultimediaItem::getCreator()

{

return creator;

}

void MultimediaItem::setPublicationYear(int py)

{

publicationYear = py;

}

int MultimediaItem::getPublicationYear()

{

return publicationYear;

}

void Book::setAuthor(string a) {

author = a;

}

string Book::getAuthor() {

return author;

}

void Book::setGenre(string g) {

genre = g;

}

string Book::getGenre() {

return genre;

}

void Book::setnumber\_of\_pages(int np) {

number\_of\_pages = np;

}

int Book::getnumber\_of\_pages() {

return number\_of\_pages;

}

void Video:: setFormat(string f) {

format = f;

}

string Video::getFormat() {

return format;

}

void Video::setDuration(int d) {

duration = d;

}

int Video::getDuration() {

return duration;

}

void Video::setResolution(int r) {

resolution = r;

}

int Video::getResolution() {

return resolution;

}

void MusicAlbum::setArtist(string ar) {

artist = ar;

}

string MusicAlbum::getArtist() {

return artist;

}

void MusicAlbum::setGenre(string g) {

genre = g;

}

string MusicAlbum::getGenre() {

return genre;

}

void MusicAlbum::setNumTracks(int nt) {

numTracks = nt;

}

int MusicAlbum::getNumTracks() {

return numTracks;

}

void displayDetails(Book b) {

cout << "Title: " << b.getTitle() << endl;

cout << "Creator: " << b.getCreator() << endl;

cout << "Publication Year: " << b.getPublicationYear() << endl;

cout << "Author: " << b.getAuthor() << endl;

cout << "Genre: " << b.getGenre() << endl;

cout << "Number of Pages: " << b.getnumber\_of\_pages() << endl;

}

void displayDetails(Video v)

{

cout << "Title: " << v.getTitle() << endl;

cout << "Creator: " << v.getCreator() << endl;

cout << "Publication Year: " << v.getPublicationYear() << endl;

cout << "Format: " << v.getFormat() << endl;

cout << "Duration: " << v.getDuration() << endl;

cout << "Resolution: " << v.getResolution() << endl;

}

void displayDetails(MusicAlbum m)

{

cout << "Title: " << m.getTitle() << endl;

cout << "Creator: " << m.getCreator() << endl;

cout << "Publication Year: " << m.getPublicationYear() << endl;

cout << "Artist: " << m.getArtist() << endl;

cout << "Genre: " << m.getGenre() << endl;

cout << "Number of Tracks: " << m.getNumTracks() << endl;

}

//MAIN file (source.cpp)

#include<iostream>

#include<string>

#include"interface2.h"

using namespace std;

int main() {

Book b("dfvfdvfdv", "fadvdfzv", 1951, "svfdag", "Lafgadfg", 277);

Video v("fdagfdag", "Fadfgdfa", 1972, "adfgad", 175, 1080);

MusicAlbum m("adfgadf", "Micadfgadhael adf", 1982, "dfa Jackdaffdagson", "afdgfda", 9);

displayDetails(b);

cout << endl;

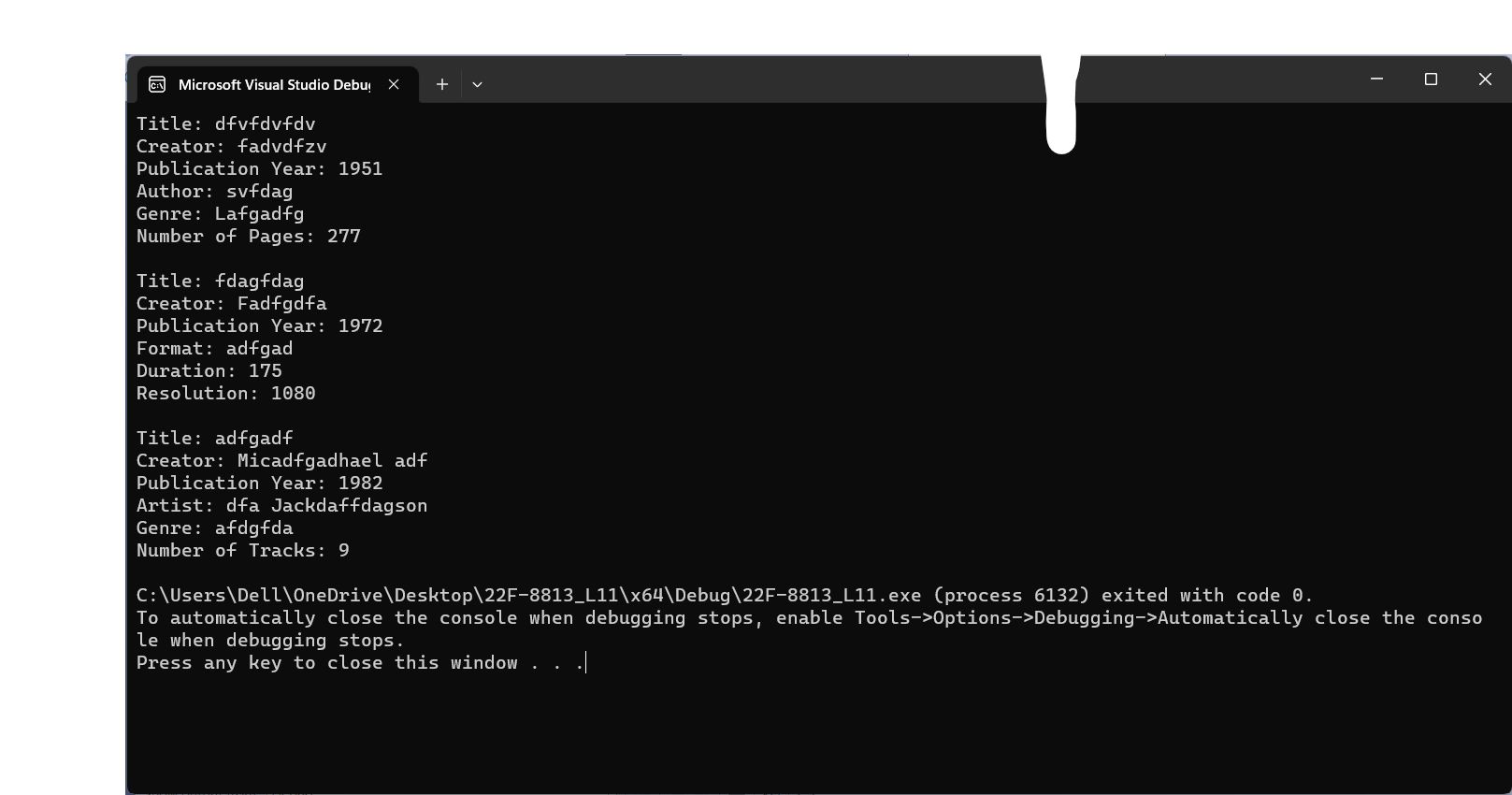
displayDetails(v);

cout << endl;

displayDetails(m);

return 0;

}



**Task 4:**

// (class header file)

#pragma once

#ifndef CLASS4\_H

#include<string>

#include<iostream>

using namespace std;

#define CLASS4\_H

#endif

class Employee {

private:

string name;

int age;

float salary;

public:

Employee() : name(""), age(0), salary(0.0) {}

Employee(string name, int age, float salary) : name(name), age(age), salary(salary) {}

friend istream& operator>>(istream& is, Employee& emp);

friend ostream& operator<<(ostream& os, const Employee& emp);

friend Employee operator+(const Employee& emp1, const Employee& emp2);

friend bool operator<(const Employee& emp1, const Employee& emp2);

friend bool operator>(const Employee& emp1, const Employee& emp2);

};

//Class.cpp file (class source file)

#include "interface4.h"

istream& operator>>(istream& is, Employee& emp) {

cout << "Enter employee name: ";

getline(is, emp.name);

cout << "Enter employee age: ";

is >> emp.age;

cout << "Enter employee salary: ";

is >> emp.salary;

return is;

}

ostream& operator<<(ostream& os, const Employee& emp) {

os << "Employee Name: " << emp.name << endl;

os << "Employee Age: " << emp.age << endl;

os << "Employee Salary: " << emp.salary << endl;

return os;

}

Employee operator+(const Employee& emp1, const Employee& emp2) {

string newName = emp1.name + " " + emp2.name;

int newAge = emp1.age + emp2.age;

float newSalary = emp1.salary + emp2.salary;

Employee emp3(newName, newAge, newSalary);

return emp3;

}

bool operator<(const Employee& emp1, const Employee& emp2) {

return emp1.salary < emp2.salary;

}

bool operator>(const Employee& emp1, const Employee& emp2) {

return emp1.salary > emp2.salary;

}

//MAIN file (source.cpp)

#include<iostream>

#include<string>

#include"interface4.h"

using namespace std;

int main()

{

Employee obj1, obj2, obj3;

cout << "Enter details for Employee 1: " << endl;

cin >> obj1;

cout << "Enter details for Employee 2: " << endl;

cin >> obj2;

obj3 = obj1 + obj2;

cout << "Details of Employee 1: " << endl;

cout << obj1 << endl;

cout << "Details of Employee 2: " << endl;

cout << obj2 << endl;

cout << "Details of Employee 3: " << endl;

cout << obj3 << endl;

if (obj1 < obj2) {

cout << "Employee 2 has higher salary." << endl;

}

else {

cout << "Employee 1 has higher salary." << endl;

}

return 0;

}

