**Lab 7**

**Name:** Etcherla Sai Manoj **Mis. No:** 112015044 **Branch:** CSE

**Question1:**

**Code:**

#include<iostream>

#include<cstring>

using namespace std;

class Amazon\_music{

public:

string singer\_name, album\_name;

int no\_songs, playtime;

Amazon\_music(){

cout << "Enter singer name : ";

cin >> singer\_name;

cout << "Enter album name : ";

cin >> album\_name;

cout << "Enter number of songs :";

cin >> no\_songs;

cout << "Enter playtime in minutes : ";

cin >> playtime;

}

virtual void display(){

cout << "Singer name : " << singer\_name << endl;

cout << "Album name : " << album\_name << endl;

cout << "Number of songs :" << no\_songs << endl;

cout << "Playtime in minutes : " << playtime << endl;

}

virtual ~Amazon\_music(){

cout << "\nAmazon\_music destructor called.....\n";

};

};

class Music\_info : public Amazon\_music{

string genre;

public:

Music\_info(){

cout << "Enter genre :";

cin >> genre;

}

virtual void display(){

Amazon\_music::display();

cout << "Genre : " << genre << endl;

}

~Music\_info(){

cout << "\nMusic\_info desturctor called.....\n";

};

};

int main(){

Amazon\_music a, \*p1, \*p2;

Music\_info m;

cout << "---------------------------------------" << endl;

a.display();

cout << "---------------------------------------" << endl;

m.display();

cout << "---------------------------------------" << endl;

p1 = &a;

p1->display();

cout << "---------------------------------------" << endl;

p2 = &m;

p2->display();

cout << "---------------------------------------" << endl;

Amazon\_music \*p = new Music\_info(m);

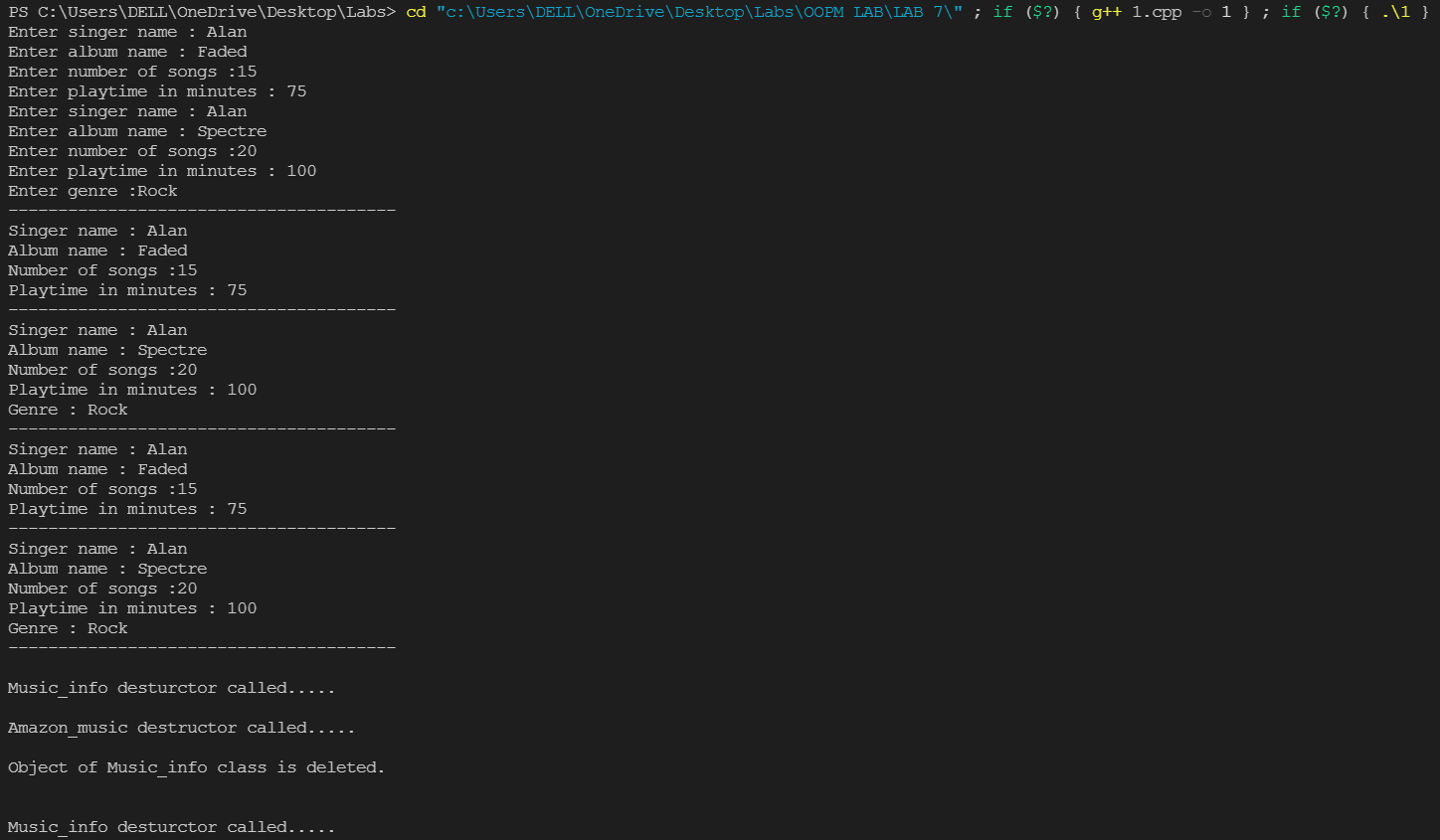
delete p;

cout << "\nObject of Music\_info class is deleted.\n" << endl;

return 0;

}

**Input & Output:**

****

****

**Question2:**

**Code:**

#include<iostream>

#include<cstring>

using namespace std;

class Account\_Details{

public:

string client\_name;

int acc\_no;

double curr\_bal;

double withdraw\_amount, deposit\_amount;

void create(){

cout << "Enter your name : ";

cin >> client\_name;

cout << "Enter your account number :";

cin >> acc\_no;

cout << "Enter your current balance :";

cin >> curr\_bal;

}

void deposit\_money(){

double deposit\_amount;

cout << "Enter amount you want to deposit : ";

cin >> deposit\_amount;

curr\_bal = curr\_bal + deposit\_amount;

cout << "Your current balance : " << curr\_bal << endl;

}

void withdraw\_money(){

double withdraw\_amount;

cout << "Enter amount you want to withdraw : ";

cin >> withdraw\_amount;

curr\_bal = curr\_bal - withdraw\_amount;

cout << "Your current balance : " << curr\_bal << endl;

}

void display(){

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*DETAILS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "Enter your name : " << client\_name << endl;

cout << "Enter your account number :" << acc\_no << endl;

cout << "Enter your current balance :" << curr\_bal << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

};

class Regular : public Account\_Details{

};

class VIP : public Account\_Details{

public:

double upper\_limit, interest, pending;

VIP(){

upper\_limit = 500;

interest = 11.125;

}

virtual void create(){

Account\_Details::create();

cout << "Enter upper-limit to loan : ";

cin >> upper\_limit;

cout << "Enter simple interest to loan : ";

cin >> interest;

}

virtual void withdraw\_money(){

int time;

cout << "Enter money you want to withdraw : ";

cin >> withdraw\_amount;

if(curr\_bal < withdraw\_amount){

if(withdraw\_amount - curr\_bal <= upper\_limit){

cout << "Enter time period to return loan : ";

cin >> time;

pending = (withdraw\_amount - curr\_bal) + ((withdraw\_amount-curr\_bal) \* time \* interest)/100;

}

else{

cout << "Difference amount is greater than Upper-limit exceeded";

}

}

else{

pending = 0;

}

curr\_bal = curr\_bal - withdraw\_amount;

}

virtual void display(){

Account\_Details::display();

cout << "Pending amount to pay : " << pending << endl;

}

};

int main(){

cout << "--------------Regular----------------\n";

Regular r;

r.create();

r.deposit\_money();

r.withdraw\_money();

r.display();

cout << "--------------------------------------\n\n";

cout << "----------------VIP-------------------\n";

VIP v;

v.create();

v.deposit\_money();

v.withdraw\_money();

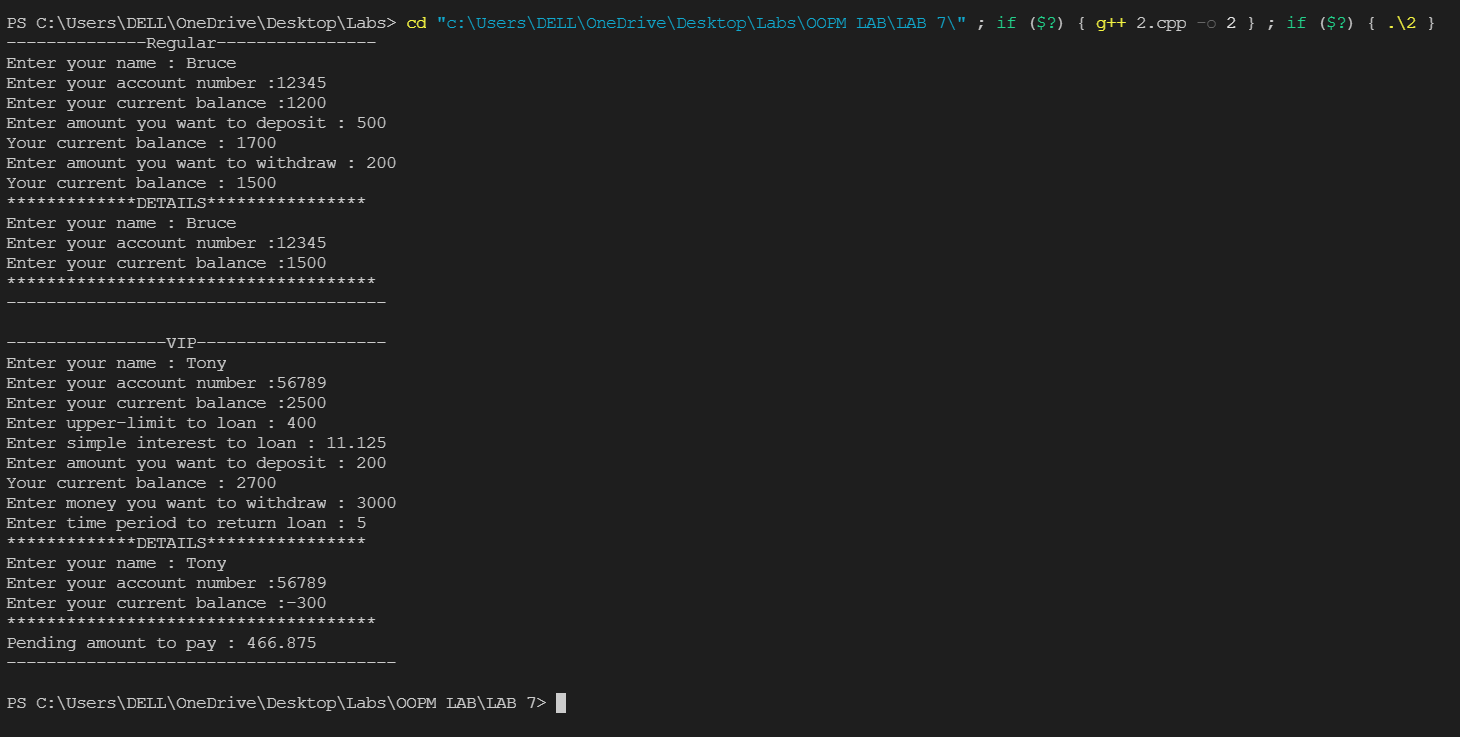
v.display();

cout << "---------------------------------------\n\n";

return 0;

}

**Input & Output:**

****