

Object Oriented Programming Lab

Lab Task 7

Submitted by: 241484\_Huzaifa Basit

To: Sir Mahaz

Cyber Security Fall 2024-B

1. Binary Class with Operator Overloading

C++ Code:

#include<iostream>

using namespace std;

class Binary

{

float num;

public:

Binary(float num = 0);

Binary operator+(Binary b);

Binary operator-(Binary b);

Binary operator\*(Binary b);

Binary operator/(Binary b);

friend istream& operator>>(istream &in, Binary &b);

friend ostream& operator<<(ostream &out, Binary &b);

};

int main()

{

Binary b1, b2, result;

cout<<"Enter first binary number: ";

cin>>b1;

cout<<"Enter second binary number: ";

cin>>b2;

result = b1 + b2;

cout<<"Addition: "<<result;

result = b1 - b2;

cout<<"Subtraction: "<<result;

result = b1 \* b2;

cout<<"Multiplication: "<<result;

result = b1 / b2;

cout<<"Division: "<<result;

}

Binary::Binary(float num)

{

this->num = num;

}

Binary Binary::operator+(Binary b)

{

return Binary(num + b.num);

}

Binary Binary::operator-(Binary b)

{

return Binary(num - b.num);

}

Binary Binary::operator\*(Binary b)

{

return Binary(num \* b.num);

}

Binary Binary::operator/(Binary b)

{

if (b.num != 0)

return Binary(num / b.num);

else

return Binary(0);

}

istream& operator>>(istream &in, Binary &b)

{

in >> b.num;

return in;

}

ostream& operator<<(ostream &out, Binary &b)

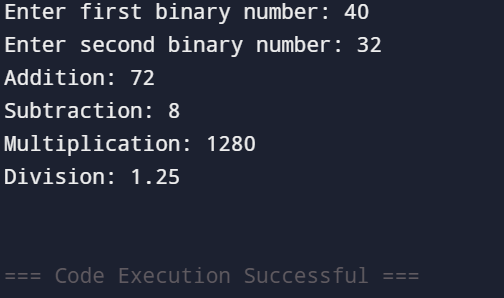
{

out << b.num << endl;

return out;

}

Output:



1. Simple Library Management System

C++Code:

#include<iostream>

using namespace std;

class Book

{

string title, author;

int year;

public:

Book(string t="Unknown", string a="Unknown", int y=0);

void setbook(string t, string a, int y);

void display();

friend class Patron;

};

class Patron

{

string name;

int id;

Book\* borrowed[10]; // Max 10 books can be borrowed

int count;

public:

Patron(string n="Unknown", int i=0);

void setpatron(string n, int i);

void showpatron();

void showborrowed();

void operator+(Book\* b); // Overload + operator to borrow book

};

class Library

{

Book\* books[100]; // Aggregation (Array of pointers to Book)

int bookcount;

public:

Library();

void addbook(Book\* b);

void showall();

};

int main()

{

Library lib;

Book\* b1 = new Book("C++ Basics", "Bjarne", 1985);

Book\* b2 = new Book("OOP Concepts", "Grady", 1994);

Book\* b3 = new Book("Data Structures", "Mark", 2000);

lib.addbook(b1);

lib.addbook(b2);

lib.addbook(b3);

Patron p1("Ali", 1);

Patron p2("Sara", 2);

p1 + b1;

p1 + b2;

p2 + b3;

cout<<"\nLibrary Books:\n";

lib.showall();

cout<<"\nPatron Details:\n";

p1.showpatron();

p1.showborrowed();

p2.showpatron();

p2.showborrowed();

}

// Book class definitions

Book::Book(string t, string a, int y)

{

title = t;

author = a;

year = y;

}

void Book::setbook(string t, string a, int y)

{

title = t;

author = a;

year = y;

}

void Book::display()

{

cout<<"Title: "<<title<<endl;

cout<<"Author: "<<author<<endl;

cout<<"Year: "<<year<<endl;

}

// Patron class definitions

Patron::Patron(string n, int i)

{

name = n;

id = i;

count = 0;

}

void Patron::setpatron(string n, int i)

{

name = n;

id = i;

}

void Patron::showpatron()

{

cout<<"Name: "<<name<<endl;

cout<<"ID: "<<id<<endl;

}

void Patron::showborrowed()

{

cout<<"Borrowed Books:\n";

for(int i=0; i<count; i++)

{

borrowed[i]->display();

cout<<endl;

}

}

void Patron::operator+(Book\* b)

{

if(count < 10)

{

borrowed[count] = b;

count++;

}

}

// Library class definitions

Library::Library()

{

bookcount = 0;

}

void Library::addbook(Book\* b)

{

books[bookcount] = b;

bookcount++;

}

void Library::showall()

{

for(int i=0; i<bookcount; i++)

{

books[i]->display();

cout<<endl;

}

}

Output:

