



Lab Report

CSE 122



**Submitted To:**

Humayra Ferdous

Lecturer

Department of CSE

Bangladesh University Of Business And Technology

**Submitted By:**

Name: Istiak Ahmed Saad

ID: 22234103008

Section: 01

Intake: 50

Bangladesh University Of Business And Technology

March 20, 2023

Bangladesh University of Business and Technology

**Experiment no-01:**

*Class Design*

Create a Class named Vehicle.

a. It must have three member variables.

b. Class must have three member functions.

I. One function would take value from the keyboard (Function name: getValue)

II. Another One’s value would be assigned by a programmer (Function name: setValue)

III. Third function would print the output (Function name: printValue)

**Main Function:**

Create two objects of your created class. One object would take value from the keyboard and another one’s value would be assigned. Print the values of two objects.

**Object:**

Demonstrate the implementation of a class named "Vehicle" that contains member variables and functions for taking input, setting values, and printing output.

**Experiment Details:**

The class "Vehicle" has three member variables: "serialnumber", "makingyear", and "model" of type int, int, and char array, respectively. The class "Vehicle" has three member functions: "getValue()" for taking input from the user, "setValue()" for setting values of the member variables, and "printValue()" for printing the member variables.

Two objects of the class "Vehicle" are created in the main function, one object takes input from the user using the "getValue()" function and the other object's values are set using the "setValue()" function. The output of both objects is printed using the "printValue()" function.

**OutPut:**

\*\*ENTER YOUR INPUT\*\*

Enter Vehicle Name: Bugatti

Enter Vehicle Model: Chiron Super Sport 300+

Enter Vehicle Manufacture Year: 2019

For Vehicle 1:

Vehicle Name: BMW

Vehicle Model: iX xDrive 50

Vehicle Manufacture Year: 2023

For Vehicle 2:

Vehicle Name: Bugatti

Vehicle Model: Chiron Super Sport 300+

Vehicle Manufacture Year: 2019

<< "For Vehicle 1: " << endl;

    v1.printValue();

    cout << endl;

    cout << "For Vehicle 2: " << endl;

    v2.printValue();

    cout << endl;

    return 0;

}

cout << "Enter Vehicle Manufacture Year: ";

        cin >> published;

        cin.ignore();

    }

 void printValue()

    {

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

        cout << "Vehicle Model: " << model << endl;

        cout << "Vehicle Manufacture Year: " << published << endl;

    }

};

int main()

{

    Vehicle v1;

    Vehicle v2;

    v1.setValue("BMW", "iX xDrive 50", 2023);

    v2.getValue();

    cout << endl

#include <iostream>

#include <string>

using namespace std;

class Vehicle

{

private:

    string name;

    string model;

    int published;

public:

    void setValue(string n, string m, int p)

    {

        name = n;

        model = m;

        published = p;

    }

    void getValue()

    {

        cout << "\*\*ENTER YOUR INPUT\*\*" << endl;

        cout << "Enter Vehicle Name: ";

        getline(cin, name);

cout << "Enter Vehicle Model: ";

        getline(cin, model);

**Discussion:**

The class "Vehicle" is a simple example of how to create a class with member variables and functions in C++. The class can be used to represent vehicles, and the member functions can be used to manipulate the data of the class.

**Experiment no-02:**

Create a **University** class with 5 member variables and 2 member functions(getData() and displayData()). Then make an array of 3 University objects and use two functions to get information and show information.

**Objective:**

The objective of this code is to create a class named University and then define functions to input and output data of university.

**Experiment Details:**

This code defines a University class with private attributes (name, location, mobilenumber, year, and numberofstudent) and public member functions (getData and displayData) to input and output data, respectively. It then creates an object array of the University class with a size of 3 and takes input for each object. Finally, it displays the data for all objects.

cout << "Your University is: " << public\_private << endl;

        cout << endl;

    }

};

int main()

{

    int i;

    University u[3];

    cout << "\*\*\*Enter Your University Information\*\*\*" << endl;

    for (i = 0; i < 3; i++)

    {

        u[i].getData();

    }

    cout << endl;

    cout << "\*\*Your Entered Data\*\*" << endl;

    for (i = 0; i < 3; i++)

cout << "Number Of Students: ";

        cin >> number\_of\_student;

        cin.ignore();

        cout << "University Location: ";

        getline(cin, location);

        cout << "Is Your University Public or Private?: ";

        getline(cin, public\_private);

    }

    void displayData()

    {

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

        cout << "University Published Year: " << founded\_year << endl;

        cout << "University Number of Students: " << number\_of\_student << endl;

        cout << "University Location: " << location << endl;

#include <iostream>

#include <string>

using namespace std;

class University

{

private:

    string name;

    int founded\_year;

    int number\_of\_student;

    string location;

    string public\_private;

public:

    void getData()

    {

        cout << "Enter University Name: ";

        getline(cin, name);

        cout << "University Published Year: ";

        cin >> founded\_year;

**Output:**

\*\*\*Enter Your University Information\*\*\*

Enter University Name: Bangladesh University of Business and Technology

University Published Year: 2002

Number Of Students: 5000

University Location: Dhaka

Is Your University Public or Private?: Private

Enter University Name: Dhabi

University Published Year: 1985

Number Of Students: 500000

University Location: Dhaka

Is Your University Public or Private?: Public

Enter University Name: Buet

University Published Year: 1995

Number Of Students: 100000

University Location: Dhaka

Is Your University Public or Private?: Public

\*\*Your Entered Data\*\*

University Name: Bangladesh University of Business and Technology

University Published Year: 2002

University Number of Students: 5000

University Location: Dhaka

Your University is: Private

University Name: Dhabi

University Published Year: 1985

University Number of Students: 500000

University Location: Dhaka

Your University is: Public

University Name: Buet

University Published Year: 1995

University Number of Students: 100000

University Location: Dhaka

Your University is: Public

{

        u[i].displayData();

    }

    return 0;

}

**Output:**

University Location: Dhaka

Your University is: Public

Discussion:

This code is a basic implementation of a class in c++. It demonstrates how to use class objects to store and manipulate data. By creating an object array of the University class, it is possible to store data for multiple universities. The member function to the University class can be used to access the data for input and output. This code can be extended to include additional functionality, such as searching and sorting universities based on their attributes.

**Experiment no-03:**

Create a class called Reverse. This class must include one integer variable, one getInput ()

function, and one doReverse() function. Your task is to take an integer input from the keyboard

and reverse it.

**Objective:**

The objective of this code is to reverse a given integer using a class named Reverse.

**Experiment Details:**

This code defines a class named Reverse with a private attribute (num) and public member functions (getInput and doReverse). The getInput function takes input from the user and stores it in the num attribute. The doReverse function reverses the given integer using a while loop and displays the reverse value.

#include <iostream>

using namespace std;

class Reverse

{

private:

int num;

public:

void getInput()

{

cout << "Enter an Integer: ";

cin >> num;

}

void doReverses()

{

int rev\_num = 0;

int temp\_num = num;

while (temp\_num > 0)

{

rev\_num = rev\_num \* 10 + (temp\_num % 10);

temp\_num /= 10;

}

cout << "The Reverse of " << num << " is " << rev\_num << endl;

}

};

int main()

{

Reverse rev;

rev.getInput();

rev.doReverses();

return 0;

}

**Output:**

Enter an Integer: 4321

The Reverse of 4321 is 1234

Discussion:

This code is a basic implementation of a class in c++ reverse a given integer. It demonstrates how to use class objects to store and manipulate data. The member functions of the Reverse class can be used to access the input data and reverse. This code can be extended to include additional functionality.

**Experiment no-04:**

A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767), and the number (8900). Write a program that uses a class (Phone) to store these three parts of a phone number separately. Create two objects of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers. The interchange might look like this: Enter your area code, exchange, and number: 415 555 1212 My

number is (212) 767-8900 Your number is (415) 555-1212

**Objective:**

Demonstrate the use of a class to represent a phone number, and the creation of phone objects.

**Experiment Details:**

The program defines a Phone class to represent a phone number with three components: area code, exchange, and number. The class has two constructors, one default, and another that takes the three components as arguments. The class has two member functions, getValue() and displayNumber(), to input and display the phone number, respectively. The main function creates two Phone objects and uses them to display the phone numbers.

**Output:**

Enter your area code, exchange code & number: 415 555 1212

My Number is (212) 767-8900

Your Number is: My Number is (415) 555-1212

void dispalyNumber()

{

cout << "My Number is (" << area\_code << ") " << exchange << "-" << number << endl;

}};

int main()

{

Phone p1(212, 767, 8900);

Phone p2(0, 0, 0);

p2.getInput();

p1.dispalyNumber();

cout << "Your Number is: ";

p2.dispalyNumber();

return 0;

}

#include <iostream>

#include <string>

using namespace std;

class Phone

{

private:

int area\_code;

int exchange;

int number;

public:

Phone(int a\_c, int e, int num)

{

area\_code = a\_c;

exchange = e;

number = num;

}

void getInput(){

cout << "Enter your area code, exchange code & number: ";

cin >> area\_code >> exchange >> number; }

Discussion:

This program demonstrates the use of a class to create objects to represent a phone number. The Phone class has three data members that hold the area code, exchange, and number of the phone. The class has two constructors, a default constructor, and another constructor that takes the three components as arguments. The getValue() function is used to input the phone number, and the displayNumber() function displays the phone number.

**Experiment no-05:**

Create a class called time. Its three members, all type int, should be called hours, minutes, and seconds. Write a program that prompts the user to enter a time value in hours, minutes, and seconds using the getTime() function. The program should print out the total number of seconds using DisplayTimeInSeconds() function.

**Objective:**

The objective of this code is to convert time in hours, minutes, and seconds into total seconds using a class named time.  
  
**Experiment Details:**

This code defines a class named time with private attributes (hours, minutes, and seconds) and public member functions (getTimes and DisplayTimeInSeconds). The getTimes function takes input from the user and stores it in the hours, minutes, and seconds attributes. The DisplayTimeInSeconds function converts the time into total seconds and displays it.

**Output:**

Enter Hours:8

Enter Minutes:48

Enter Seconds:23

Input Total Time In Seconds:31703

cin >> seconds;

}

void DisplayTimeInSeconds()

{

int sec = (hours \* 3600) + (minutes \* 60) + seconds;

cout << "Input Total Time In Seconds:" << sec << endl;

}

};

int main()

{

time d;

d.getTime();

d.DisplayTimeInSeconds();

return 0;

}

#include <iostream>

using namespace std;

class time

{

private:

int hours;

int minutes;

int seconds;

public:

void getTime()

{

cout << "Enter Hours:";

cin >> hours;

cout << "Enter Minutes:";

cin >> minutes;

cout << "Enter Seconds:";

cin >> seconds;

}

Discussion:  
 This code is a basic im implementation of a class

In c++ to convert time hours, minutes, and second into total seconds. It demonstrates how to use class objects to store and manipulate data. The member functions of the time class can be used to access the input data.

**Experiment no-06:**

Find out which student got the highest mark among the 5 students. For this, you have to create a

Student class. In this class,

Pass an array of objects as an argument in a function.

Create necessary constructors and functions.

**Objective:**

The objective of this code is to find the student with the highest marks among 5 students using classes and objects in C++.

**Experiment Details:**

The program defines a class called "Student" that has three private data members: id, name, and mark. The class has three member functions: getInput() to take input for id, name, and mark, Check() to compare the marks of two students, and Display() to display the student's details with the highest mark. The main function creates an array of 5 Student objects, takes input for each student, and then finds the student with the highest mark using the Check() function and displays the result using the Display() function.

Discussion:

**Output:**

Student Get Highest Mark is Tajrin with a mark of 87

}

cout << "Student Get Highest Mark is " << s1[highest].getName() << " with a mark of " << s1[highest].getMark() << endl;

}

int main()

{

Student s1[5];

s1[0] = Student("Saad", 1, 80.0);

s1[1] = Student("Eimran", 2, 82.0);

s1[2] = Student("Tajrin", 3, 87.0);

s1[3] = Student("Raya", 4, 85.0);

s1[4] = Student("Bonhy", 5, 86.0);

highestMark(s1);

return 0;

}

#include <iostream>

#include <string>

using namespace std;

class Student

{

private:

string name;

int roll;

float mark;

public:

Student(string n, int r, float m)

{

name = n;

roll = r;

mark = m;

}

Student()

{ name = "";roll = 0;

mark = 0;

}

string getName()

{

return name;

}

int getRoll()

{

return roll;

}

float getMark()

{

return mark;

}

};

void highestMark(Student s1[])

{

int highest = 0, i;

for (i = 0; i < 5; i++)

{

if (s1[i].getMark() > s1[highest].getMark())

{

highest = i;

}

This program demonstrates the use of classes and objects in C++ to implement a real-world problem. The program allows the user to input data for 5 students and then finds the student with the highest mark using a class method. The code can be useful in a situation where we want to find the student with the highest mark in a class or in a school. However, the program assumes that there is only one student with the highest mark, and it does not consider the possibility of multiple students having the same highest mark.

**Experiment no-07:**

Create a class named time that must have three integer data members (hours, minutes, and

seconds). Create two constructors, one member function named add\_time (return object), and one

display function to print the time in 11:59:59 format. The add\_time function must take two objects

as arguments. The main function calls the add\_time function to add two-time objects and store the

result in a third object. Use the display function to print the result on the console.

Input: 2 55 40, 5 20 30

Output: 8:16:10

**Objective:**

The objective of this code is to create a class for time and define methods to add two times.

**Experiment Details:**

The code defines a class "time" with constructors and methods to add two time objects. The "add\_time" method takes another time object as input and returns a new time object with the sum of the two times. The "display" method displays the time in hours, minutes, and seconds format.

Discussion:  
The code is well-written and demonstrates the use of classes and methods to perform time addition. It also shows the concept of constructor overloading. Overall, it is a good example of object-oriented programming in C++.

**Output:**

TIME FORMAT:8:16:10

{

time t3;

t3.seconds = seconds + t1.seconds + t2.seconds - 60;

t3.minutes = minutes + t1.minutes + t2.minutes - 60 + 1;

t3.hours = hours + t1.hours + t2.hours + 1;

return t3;

}

void display()

{

cout << "TIME FORMAT:" << hours << ":" << minutes << ":" << seconds << endl;

}};

int main()

{

time t1(2, 55, 40);

time t2(5, 20, 30);

time t3 = t3.add\_time(t1, t2);

t3.display();

return 0;

}

#include <iostream>

using namespace std;

class time

{

private:

int hours;

int minutes;

int seconds;

public:

time()

{

hours = 0;

minutes = 0;

seconds = 0;

}

time(int a, int b, int c)

{

hours = a;

minutes = b;

seconds = c;

}time add\_time(time t1, time t2)

**Experiment no-08:**

Create a class named Complex that must have two integer data members (real, and imag). Create two constructors, one Read function to take keyboard input, one Add (return object), function, and one Display function to print results. The Add function must take one object as an argument. The task of the main function is the same as question 2.

Input

Enter real and imaginary numbers respectively: 16 7

Enter real and imaginary numbers respectively: 5 8

Output:

Sum = 21 + 15i

**Objective:**

To create a program that performs addition of two complex numbers and displays the result.

**Experiment Details:**

The program defines a class called Complex that contains two private data members - real and imaginary - to represent the real and imaginary parts of a complex number. The class also includes two constructor functions, one default and one parameterized, an input function to read values for the real and imaginary parts of the complex numbers, an Add function to add two complex numbers, and a Display function to display the result.

Discussion:

**Output:**

Enter real and imaginary number respectively:16 7

Enter real and imaginary number respectively:5 8

SUM=21+15i

complex add(complex c1)

{

complex c3;

c3.real = real + c1.real;

c3.imag = imag + c1.imag;

return c3;

}void display()

{

cout << "SUM=" << real << "+" << imag << "i" << endl;

}};

int main()

{

complex c1, c2, c3;

c1.read();

c2.read();

c3 = c1.add(c2);

c3.display();

return 0;

}

#include <iostream>

using namespace std;

class complex

{

private:

int real;

int imag;

public:

complex()

{

real = 0;

imag = 3;

}complex(int a, int b)

{

real = a;

imag = b;

}void read()

{

cout << "Enter real and imaginary number respectively:";

cin >> real >> imag;

}

The program uses a class-based approach to perform addition of two complex numbers. It takes input for the real and imaginary parts of the two complex numbers, creates two objects of the Complex class, and calls the Add function on one of the objects, passing the other object as a parameter. The Add function performs the addition operation and returns the result as a new Complex object, which is then displayed using the Display function.

**Experiment no-09:**

Create a class for the above Triangle.

a. Create one default constructor.

b. Create four parameterized constructors. They would look like this –



Constructor\_name(double, double)

Constructor\_name(double, int)

Constructor\_name(int, double)

Constructor\_name(int, int)

c. Create one copy constructor.

d. Create a function that would return the area of the triangle

e. Create a destructor.

Call each of the functions from the main functions. Call Constructor\_name(double, int) using

pass-by-value and Constructor\_name(int, double) by pass-by-reference.

**Objective:**

The objective of this experiment is to create a class named Triangle in C++ programming language that can calculate the area of a triangle based on its height and base.

**Experiment Details:**

In this experiment, we will create a class named Triangle in C++ programming language with various constructors that can take different types of input parameters. The class will also have a member function named area that will calculate the area of a triangle based on its height and base.

In the main function, we will create two objects of the Triangle class using different constructors and calculate their respective areas using the area member function. Finally, we will print the area of both triangles on the console.

Discussion:

**Output:**

Area Of Triangle:9.6

Invoked Destructor

Invoked Destructor

Invoked Destructor

Invoked Destructor

Invoked Destructor

Invoked Destructor

double &ptr2 = y;

Triangle s4(ptr1, ptr2);

Triangle s5(7, 8);

Triangle A;

double D = A.Area(4.8, 4);

cout << "Area Of Triangle:" << D << endl;

return 0;

}

#include <iostream>

using namespace std;

class Triangle

{

private:

double H, B;

public:

Triangle()

{

H = 11.3;

B = 8.7;

}Triangle(double a, double b)

{

H = a;

B = b;

}Triangle(double a, int b)

{

H = a;

B = b;

}Triangle(int a, double b)

{

H = a;

B = b;

}

Triangle(const Triangle &s)

{

H = s.H;

B = s.B;

}double Area(double a, int b)

{

return 0.5 \* a \* b;

}void show()

{

double area = 0.5 \* H \* B;

cout << "Area of Triangle:" << area << endl;

}~Triangle()

{

cout << "Invoked Destructor" << endl;

}

};int main()

{

Triangle s1;

Triangle s2(4.5, 4.4);

Triangle s3(3.3, 3);

int x = 5; double y = 5.6;

int &ptr1 = x;

The given C++ program creates a class named Triangle that can calculate the area of a triangle based on its height and base. The class has various constructors that can take different types of input parameters. The program creates two objects of the Triangle class using different constructors and calculates their respective areas using the area member function. Finally, the program prints the area of both triangles on the console. Overall, the program demonstrates the use of constructors and member functions in C++ programming language.

**Experiment no-10:**

K went to a shopping mall. He bought 6 items. At the cash counter, the cashier said if his total

purchase was greater than or equal to 500 Taka, he would get a 20% discount. Otherwise, he would

**Objective:**

The objective of this experiment is to create a class named Shopping in C++ programming language that can perform various operations such as adding the price of items, calculating the final price after applying discounts, and calculating the change.

**Experiment Details:**

In this experiment, we will create a class named Shopping in C++ programming language with various member functions such as amount, calculateAfterDiscount, and calculateChange. The amount function will add the price of items, the calculateAfterDiscount function will calculate the final price after applying discounts, and the calculateChange function will calculate the change.

In the main function, we will create several objects of the Shopping class and perform various operations such as adding the prices, calculating the final price after applying discounts, and calculating the change. Finally, we will print the change amount on the console.

Discussion:

**Output:**

Change 532.

C=c;

D=d;

E=e;

F=f;

}void getDiscount()

{

int sum=A+B+C+D+E+F;

if(sum>=500)

{

int returnmoney=1000-(sum-(sum\*20)/100);

cout<<returnmoney;

}else

{

int returnmoney=1000-(sum-(sum\*2)/100);

cout<<returnmoney;

}}};

int main()

{

Discount d1;

Discount d2(73,192,85,75,86,74);

d2.getDiscount();

return 0;}

#include<iostream>

using namespace std;

class Discount

{

int A;

int B;

int C;

int D;

int E;

int F;

public:

Discount()

{

A=0;

B=0;

C=0;

D=0;

E=0;

F=0;

}Discount(int a,int b,int c,int d,int e,int f)

{

A=a;

B=b;

C=c;

D=d;

E=e;

F=f;

}

The given C++ program creates a class named Shopping that performs various operations related to shopping such as adding prices, calculating the final price after applying discounts, and calculating the change. The program uses various member functions of the Shopping class to perform these operations. Finally, the program prints the change amount on the console. Overall, the program demonstrates the use of classes and member functions in C++ programming language.