# EXPERIMENT NUMBER –Practical 3.1

STUDENT’S NAME – Shinde Smita Shahaji

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

# TOPIC OF EXPERIMENT –

# Write a program to find the largest& smallest of three numbers. (Use inline function MAX and MIN)

# AIM OF THE EXPERIMENT –

# 

# Learn how to implement Object Oriented Concepts like inline and static data members in C++ Programming.

# FLOWCHART/ ALGORITHM

# START

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

# Step 2 → Using comment -macro/inline function to get max of 3 numbers.

# 

# Step 3 → Defining out of three numbers which one is greater out of them.

# MAX (a, b, c) (a > b && a > c? a: (b > c? b : c))

# Step 4 → Using comment -macro/inline function to get min of 3 numbers.

# Step 5 → Defining out of three numbers which one is minimum out of them.

# MIN(a,b,c) (a < b && a < c ? a : (b < c ? b : c))

# Step 6→ Print the message and accept the input from the user.

# Step 7 → Using operator greater than (>) and Using operator greater than (>)

# Step 8 → max and min number printed on screen

# Step 9 → End the program by returning an integer valve.

# Stop.

# PROGRAM CODE -

# #include <stdio.h>

# // macro/inline function to get max of 3 numbers

# #define MAX(a,b,c) (a > b && a > c ? a : (b > c ? b : c))

# // macro/inline function to get min of 3 numbers

# #define MIN(a,b,c) (a < b && a < c ? a : (b < c ? b : c))

# int main()

# {

# int x, y, z, large, small;

# // accept 3 numbers from console

# printf("Enter 3 numbers: ");

# scanf("%d%d%d", &x, &y, &z);

# // call inline function to get the max and min of input numbers

# large = MAX(x, y, z);

# small = MIN(x, y, z);

# // print the largest and smallest numbers

# printf("\nMax between %d, %d, and %d is %d.", x, y, z, large);

# printf("\nMin between %d, %d, and %d is %d.", x, y, z, small);

# return 0;

# }

# 

# ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

# (Kindly note down the compile time errors encountered)

# Semicolons was missing.

# Parenthesis(brackets) was missing.

# Double quotes were missing in string.

# Missing space between “using namespace std”

# PROGRAMS’ EXPLANATION (in brief)-

1. **we created a inline function as MAX and MIN for getting the lowest and largest number as inline int MAX(int &a,int &b,int &c).**
2. **inline int MIN(int &a,int &b,int &c) respectively and defining a int big and storing the value of big as to find the the largest and smallest value as big =(a > b && a > c ? a : (b > c ? b : c));**
3. **for largest and big =(a < b && a < c ? a : (b < c ? b : c)); as smallest and returning big , after that in int main() function we declare int x,y,z,small,large and take a three number from user and stored in int x ,int y and int z respectively.**
4. **then for declaring largest number we store largest number in int large by using inline function as MAX and passing the value of x,y,z for detecting the largest number as (MAX(x,y,z) ) and following same process we store small number in int small by inline function as MIN and passing the value of x,y,z for detecting the smallest number as (MIN(x,y,z) ) after the showing the value of largest number and lowest number respectively .**

# In the given program we are calculating out of given input of three numbers which one is maximum and which one is minimum out of three using greater (>) than and less (<) than operator. After compering we got one min and one max number as result.

# OUTPUT

# 

# EXPRIMENT NUMBER –Practical 3.2

STUDENT’S NAME – Shinde Smita Shahaji

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

# TOPIC OF EXPERIMENT –

# A dining hall can accommodate only 50 guests. Create a class to store seat number (Generated Automatically) and name of the guests who are seated on first come first seated basis. Define functions to display name of all guests along with seat number. Write a program to show the working of this class using the concept of static data member and static function

# AIM OF THE EXPERIMENT –

# Learn how to implement Object Oriented Concepts like inline and static data members in C++ Programming.

# FLOWCHART/ ALGORITHM- START

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

# Step 2 → After that we create a class Guest than we declare guest name and seat number.

# Step 3 → Then we define of Hall class having function to allots and list guest and static member data then we allot guest all size.

# Step 4 → Than we use static member function for allotting the seat to the guest in FIFO order.

# Step 5 → We use cout to print the detail and cin for taking input from the user.

# Step 6→ We use static member function to list the guests with name and seat number and we use while loop for checking condition if condition is true then program will be run after that initialize the static member data.

# Step 7 → we call main function for allotting the seat to the guest then listing the guest with seat number will be print.

# Step 8→ Print the message and accept the input from the user.

# Step 9 → End the program by returning name of the guest and their seat number where they are seating .

# Stop.

# PROGRAM CODE-

**#include <iostream>**

**#define MAX\_SIZE 50**

**using namespace std;**

**class Guest {**

**public:**

**char name[50];**

**int seatno;**

**int alloteseat;**

**};**

**class Hall {**

**public:**

**static int front, rear,alloteseat;**

**static Guest allGuest[MAX\_SIZE];**

**static int alloteSeat() {**

**if (rear == (MAX\_SIZE - 1)) {**

**cout<< "Hall is full!";**

**return 0;**

**}**

**rear++;**

**cout<< "Enter Guest Name: ";**

**cin>>allGuest[rear].name;**

**allGuest[rear].seatno = rear + 1;**

**return 1;**

**}**

**static void listGuest() {**

**while(++front <= rear) {**

**cout<< "\nGuest " <<allGuest[front].name << " is seated on seat S" <<allGuest[front].seatno<< ".";**

**}**

**rear = front = -1;**

**}**

**};**

**int Hall::front = -1;**

**int Hall::rear = -1;**

**Guest Hall::allGuest[MAX\_SIZE] = {};**

**int main()**

**{**

**Hall::alloteSeat();**

**Hall::alloteSeat();**

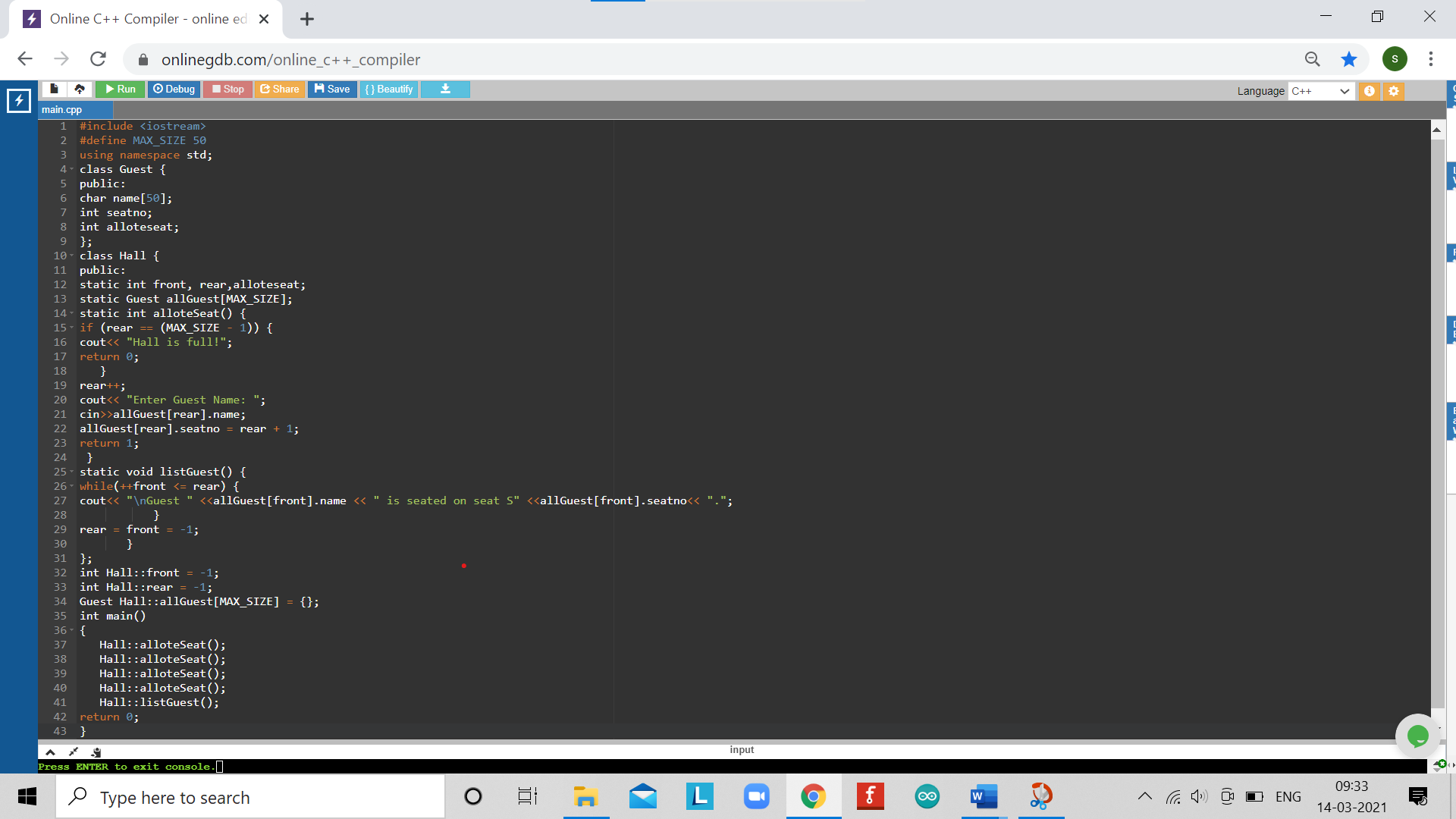
**Hall::alloteSeat();**

**Hall::alloteSeat();**

**Hall::listGuest();**

**return 0;**

**}**

****

# ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

# (Kindly note down the compile time errors encountered)

# [Error] 'intseatno does not name a type

# [Error 'staticint does not name a type In member function 'int Hall-staticintalloteSeat()':

# [Error] rear was not declared in this scope: (Error] rear was not declared in this scope

# In static member function 'static void Hallelist Guest()':

# [Error] round was not declared in this scope

# [Error] rear was not declared in this scope

# Error rear was not declared in this scope

# (Error] front was not declared in this scope

# At global scope:

# [Error] int Has front' is not a static member of 'class Hall [Error] int Hall Tear is not a static member of class Hall

# In function 'int main()':

# (Error] 'alloteSest is not a member of Hall [Error] alloteSeat is not a member of Hall

# [Error] 'altoteSeat' is not a member of 'Hall

# 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. **After that we create a class Guest than we declare guest name and seat number then we define of Hall class having function to allots and list guest and static member data then we allot guest all size.**
3. **than we use static member function for allotting the seat to the guest in FIFO order then we use cout to print the detail and cin for taking input from the user, then we use static member function to list the guests with name and seat number and we use while loop for checking condition if condition is true then program will be run after that initlizing the static member data, then we call main function for allotting the seat to the guest then listing the guest with seat number will be print.**

# OUTPUT

# 

# EXPERIMENT NUMBER –Practical 3.3

STUDENT’S NAME – Shinde Smita Shahaji

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

# TOPIC OF EXPERIMENT –

# WAP to swap private data members of classes named as class\_1, class\_2 using friend function.

# AIM OF THE EXPERIMENT –

# Learn how to implement Object Oriented Concepts like inline and static data members in C++ Programming.

# FLOWCHART/ ALGORITHM-

# START

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

# Step 2 → Declaration of function that returns integer value.

# Step 3→ Declaration of variables in integer and float datatype.

# Step 4 → Print the message and accept the input from the user.

# Step 5→ Declaration of valve of class\_1. using num variable

# Step 6→ Declaration of valve of class\_2. using num variable

# Step 7→ swapping of valves of class\_1 and class\_2

# Step 8 → End the program by returning an integer valve.

# Stop

# PROGRAM CODE

# #include<iostream>

# using namespace std;

# class class\_2;

# // defining class 1 having friend function swap

# class class\_1

# {

# // member data

# protected:

# int num1;

# public:

# class\_1()

# {

# num1=10;

# }

# // member function to show the value of member data

# void show()

# {

# cout<<"\n Value of Class 1 : "<<num1;

# }

# // friend function declaration

# friend void swap(class\_1 \*num1, class\_2 \*num2);

# };

# // defining class 2 having friend function swap

# class class\_2

# {

# // member data

# protected:

# int num2;

# public:

# class\_2()

# {

# num2=20;

# }

# // member function to show the value of member data

# void show()

# {

# cout<<"\n Value of Class 2 : "<<num2;

# }

# // friend function declaration

# friend void swap(class\_1 \*num1, class\_2 \*num2);

# };

# // definition of swap friend function

# void swap(class\_1 \*no1, class\_2 \*no2)

# {

# int no3;

# no3=no1->num1;

# no1->num1=no2->num2;

# no2->num2=no3;

# }

# int main()

# {

# class\_1 a;

# class\_2 b;

# cout<< "Values befor Swap";

# a.show();

# b.show();

# swap(&a, &b);

# cout<< "\n\nValues after Swap";

# a.show();

# b.show();

# return 0;

# }

# 

# 

# ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

# (Kindly note down the compile time errors encountered)

# Semicolons was missing.

# Parenthesis(brackets) was missing.

# Double quotes were missing in string.

# Missing space between “using namespace std;”

# PROGRAMS’ EXPLANATION (in brief)

# 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

# After that we create a class class\_2 then defining class 1 having friend function swap using member data for swapping firstly initialize num1 then member function to show the value of member data then we use cout for print the value of class 1 in num1; after that friend function declaration after that swapping for defining class 2 having friend function swap

# class class\_2 , in member data we initialize num2; then member function to show the value of member data and we use cout for print the value of class 2 in num2; then friend function will be declared . after that we define swap friend function then we use void swap for swapping class1 num1 into class 2 num 2 then we call main function and swap(&a, &b) and cout use for print the value of after swapping value will be print.

# In the given program we are calculating swap ****private data members of classes named as class\_1, class\_2 using friend function.**** Before the swapping valve of class­\_1 is 200 and class \_2 having 300. After the swapping class 1 is 300 and class 2 200 is output printed on the screen.

# OUTPUT

# EXPERIMENT NUMBER –Practical 3.4

STUDENT’S NAME – Shinde Smita Shahaji

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

# TOPIC OF EXPERIMENT –

# WAP to create a class complex to represent complex numbers. The complex class should use a function to add two complex number which are passed as arguments. The function should return an object of type complex representing the sum of two complex numbers.

# AIM OF THE EXPERIMENT –

# Learn how to implement Object Oriented Concepts like inline and static data members in C++ Programming.

# FLOWCHART/ ALGORITHM –START

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

# Step 2 → Declaration of function that returns integer value.

# Step 3→ Declaration of variables in integer and float datatype.

# Step 4→ Declaration of valve of class for complex number

# Step 5→Print the message and accept the input from the user

# Step 6→ adding (sum) of two complex number

# Step 7→ using real and float sum of complex as result.

# Step 8 → End the program by returning an integer valve.

# Stop

# PROGRAM CODE ­-

# #include<iostream>

# using namespace std;

# // defining a class to represent a complex number

# class complex

# {

# // member data to store the complex number parts

# private:

# // real part

# float r;

# // imaginary part

# float i;

# public:

# // set the values

# void set(float real, float img)

# {

# r = real;

# i = img;

# }

# // member function to sum the self and one another complex number

# complex sum(complex c)

# {

# complex t;

# t.r = r + c.r;

# t.i = i + c.i;

# return t;

# }

# // function to print the complex number

# void disp()

# {

# // since the imaginary number multiplicant of some real number

# // and thus when 1 is multiplied with any number will remain same

# if (i == -1) {

# cout<< r << " + -i" <<endl;

# }

# else if (i == 1) {

# cout<< r << " + i" <<endl;

# }

# else if (i == 0) {

# // since imaginary part is zero so only real part will be available

# cout<< r <<endl;

# }

# else {

# cout<< r << " + " <<i<< "i" <<endl;

# }

# }

# };

# int main()

# {

# complex c1, c2, c3;

# c1.set(9.5, 8.5);

# c2.set(7.5, 9.5);

# c3 = c1.sum(c2);

# cout<<"Complex Number 1 = ";

# c1.disp();

# cout<<"Complex Number 2 = ";

# c2.disp();

# cout<<"Complex Number 3 = ";

# c3.disp();

# return 0;

# }

# 

# 

# ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

# (Kindly Note down the compile time errors encountered)

# Semicolons was missing.

# Parenthesis(brackets) was missing.

# Double quotes were missing in string.

# Missing space between “using namespace std”

# PROGRAMS’ EXPLANATION (in brief)

# 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

# firstly we define a class to represent a complex number and use member data to store the complex number parts and then initialize real part and imaginary part then set the value and initialize the member function to sum the self and one another complex number complex sum(complex c) then we use function to print the complex number then the imaginary number multiplicand of some real number and thus when 1 is multiplied with any number will remain same after that when imaginary part is zero so only real part will be available the we call main function after that complex part value will be displayed.

# 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 | MaximumMarks | MarksObtained |
| 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 |  |