# EXPERIMENT NUMBER –Practical 2.1

STUDENT’S NAME – SHINDE SMITA SHAHAJI

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

# TOPIC OF EXPERIMENT –

# WAP to input a matrix of dimension m\*n. If base address is 1000. Find the address of (m-1, n-1) element of the matrix.

# AIM OF EXPERIMENT –

# Learn how to use looping constructs using C++

FLOWCHART/ ALGORITHM

# START

# Step 1- Declaring the variables in main function.

# Step 2 → Taking number of rows and number of columns as input from the user.

# Step 3 → Using for loops to enter the elements (rows and columns) of matrix as input from user.

# 

# Step 4→ Taking inputs of base address and size of an array element.

# Step 5→ Now finding the address of (m-1,n-1) element using formula bs+size\*(n\*(i-lr)+(j-lc))

# 

# Step 6→ Displaying the output.

# STOP

# PROGRAM CODE-

# #include<iostream>

# using namespace std;

# int main()

# {

# int b,i,j,w,lr=0,lc=0,n,m;

# int a[10][10];

# cout<<"enter the no. of rows in matrix";

# cin>>m;

# cout<<"enter no. of columns in matrix";

# cin>>n;

# cout<<"enter the elements in matrix";

# for(i=0;i<m;i++)

# {

# for(j=0;j<n;j++)

# {

# cin>>a[i][j];

# }

# }

# cout<<"enter the base address";

# 

# cin>>b;

# cout<<"enter the storage size of one element stored in array";

# cin>>w;

# i=m-1;

# j=n-1;

# cout<<"address of A[i][j]"<<b + w\*(n\*(i-lr)+(j-lc));

# return 0;

# }

ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

(Kindly note down the compile time errors encountered)

1. Semicolons was missing.
2. Parenthesis(brackets) was missing.
3. Double quotes were missing in string.
4. The variable was used without declaring it in main function

# PROGRAMS’ EXPLANATION (in brief)

# In the given program we are taking number of rows and number of columns as input from the user and using for loops to enter the elements (rows and columns) and taking inputs of base address and size of an array element. finding the address of (m-1,n-1) element using formula

# OUTPUT

Graphical user interface, text

Description automatically generated

# EXPERIMENT NUMBER –Practical 2.2

STUDENT’S NAME – SHINDE SMITA SHAHAJI

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

# TOPIC OF EXPERIMENT –

# Create a class called employee with the following details as variables within it.

# AIM OF THE EXPERIMENT –

# Learn how to use looping constructs using C++

# FLOWCHART/ ALGORITHM -

# START

# Step 1- Creating a user define datatype (class) employee which will store the information regarding Employee.

# 1. Name of the employee (string)

# 2. Age (int)

# 3. Designation (string)

# 4. Salary (double)

# Step 2→ The declaration of the data type is -.

# class Employee

# {

# char Name[25];

# int Age;

# char Desg[8];

# long Salary;

# public:

# void GetData();

# void PutData();

# };

# Step 3→ Now declaring two member functions getdata() and putdata() within class .

# Step 4 → Defining getdata and showdata in public.

# Step 5 →Making object e of class employee and using loop till entered number by user.

# Step 6 → Calling the getdata() and showdata () in main function through object e of class employee.

# Step 7 → Displaying the output.

# STOP.

# PROGRAM CODE-

# #include<iostream>

# using namespace std;

# class Employee

# {

# char Name[25];

# int Age;

# char Desg[8];

# long Salary;

# public:

# void GetData();

# void PutData();

# };

# void Employee :: GetData() //Statement 1 : Defining GetData()

# {

# cout<<"\n\tEnter Employee Name : ";

# cin>>Name;

# cout<<"\n\tEnter Employee Age : ";

# cin>>Age;

# cout<<"\n\tEnter Employee Designation:";

# cin>>Desg;

# cout<<"\n\tEnter Employee Salary : ";

# cin>>Salary;

# }

# void Employee :: PutData() //Statement 2 : Defining PutData()

# {

# cout<<"\nEmployee Name : "<<Name;

# cout<<"\nEmployee Age : "<<Age;

# cout<<"\nEmployee Designation:"<<Desg;

# cout<<"\nEmployee Salary : "<<Salary;

# }

# int main()

# {

# Employee E[2]; //Statement 3 : Creating Object

# int i;

# for(i=0;i<2;i++)

# {

# E[i].GetData(); //Statement 4 : Calling GetData()

# }

# for(i=0;i<2;i++)

# {

# E[i].PutData(); //Statement 5 : Calling PutData()

# }

# return 0;

# }

ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

(Kindly note down the compile time errors encountered)

1. Semicolons was missing.
2. Parenthesis(brackets) was missing.
3. Double quotes were missing in string.
4. The variable was used without declaring it in main function

PROGRAMS’ EXPLANATION (in brief)-

- In the given program, we have declared two member functions getdata() and putdata() within class and defining them in public . An object e of Employee class. Calling getdata ()and putdata() member function through the object E of Employee class and displaying the output.

OUTPUT

Graphical user interface, text, application

Description automatically generated

# EXPERIMENT NUMBER –Practical 2.3

STUDENT’S NAME – SHINDE SMITA SHAHAJI

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

# TOPIC OF EXPERIMENT –

# WAP to illustrate the use of scope resolution operator. Display the various values of the same variables declared at different scope levels.

# AIM OF THE EXPERIMENT –

# Learn how to use looping constructs using C++

# FLOWCHART/ ALGORITHM

# START

# Step 1→ Declare a variable outside the main function (i.e., global variable )

# Step 2 → Declare the same variable inside main function.

# Step 3 → Call the variable once by using scope resolution operator and once without using it.

# Step 4 → Print the output.

# STOP

# PROGRAM CODE

-

# #include<iostream>

# using namespace std;

# int my\_variable = 10; // Global x

# int main()

# {

# int my\_variable = 100; // Local x

# cout << "Value of global my\_variable is " << ::my\_variable<<endl;

# cout << "Value of local my\_variable is " << my\_variable<<endl;

# return 0;

# }

ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

(Kindly note down the compile time errors encountered)

1. Semicolons was missing.
2. Parenthesis(brackets) was missing.
3. Double quotes were missing in string.
4. The variable was used without declaring it in main function

PROGRAMS’ EXPLANATION (in brief)

The program initializes a variable once globally outside the main and once locally inside the main and then calls the same variable for displaying the output and difference in the results is obtained.

OUTPUT

A screenshot of a computer

Description automatically generated with medium confidence

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** | **Maximum**  **Marks** | **Marks**  **Obtained** |
| 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 |  |