EXPERIMENT 1.2

STUDENT NAME –
CLASS/GROUP-
SEMESTER- 2ND
SUB= OBJECT ORIENTED PROGRAMING WITH C++

UID-
DATE –

1)AIM OF THE EXPERIMENT –

Practical 2.1 : 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.

Practical 2.2 : Create a class called employee with the following details as variables within it.

Practical 2.3 : WAP to illustrate the use of scope resolution operator. Display the various values of the same variables declared at different scope levels.

2)ALGORITHM:

Practical 2.1: 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.

- Start
- Declare int variable, “rows“ and “columns”
- Take input of rows and columns
- Declare and define the 2D matrix, its base address and position on row and column
- Use the formula “address = base_address + size_of_each_int * (toFindRow * columns + toFindColumn)” to find the address of matrix[toFindRow][toFindColumn]
- End

Practical 2.2 : Create a class called employee with the following details as variables within it.

- Start
- Create a class “Employee” and inside public access specifier define the required variables and declare functions
- Outside of class, define the functions
- Inside main function create the employee object
- Take input by using loop and the functions
- Display the information
- End

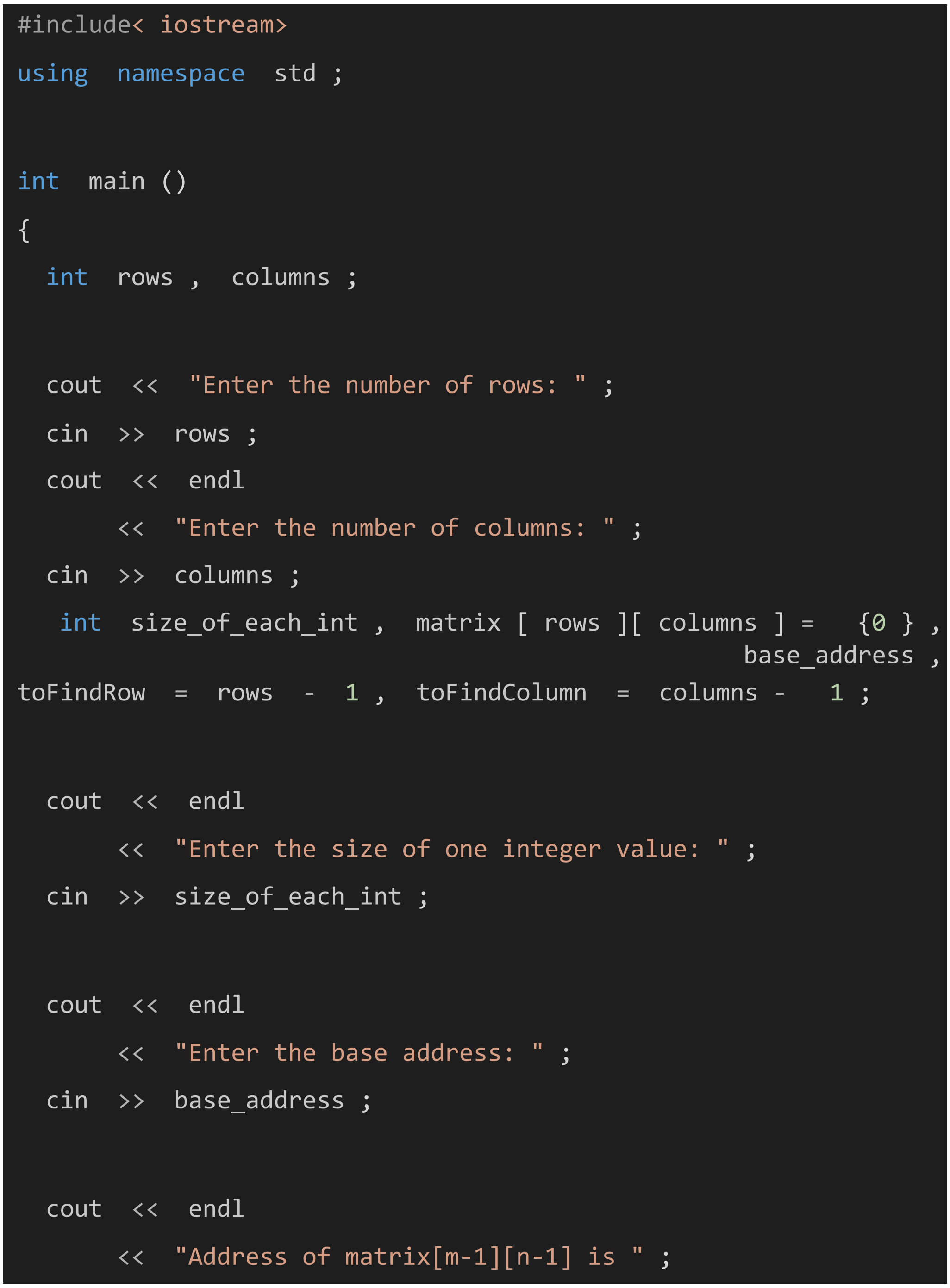
Practical 2.3 : WAP to illustrate the use of scope resolution operator. Display the various values of the same variables declared at different scope levels.

- Start

- Declare and initialize a variable inside and outside the main function
- Use “::” to access global variable and vice versa
- Show the desired output
- End

3)PROGRAM CODE

Practical 2.1: 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.



```
#include< iostream>
using  namespace  std ;


int  main ()
{
  int  rows ,  columns ;


  cout  <<  "Enter the number of rows: " ;
  cin  >>  rows ;
  cout  <<  endl
       <<  "Enter the number of columns: " ;
  cin  >>  columns ;
   int  size_of_each_int ,  matrix [ rows ][ columns ] =   {0 } ,
                                                  base_address ,
toFindRow  =  rows  -  1 ,  toFindColumn  =  columns -   1 ;


  cout  <<  endl
       <<  "Enter the size of one integer value: " ;
  cin  >>  size_of_each_int ;


  cout  <<  endl
       <<  "Enter the base address: " ;
  cin  >>  base_address ;


  cout  <<  endl
       <<  "Address of matrix[m-1][n-1] is " ;
   cout  <<  base_address  +  size_of_each_int  *  ( toFindRow  *
                                                       columns  +
toFindColumn) ;


  return  0 ;
 }
```

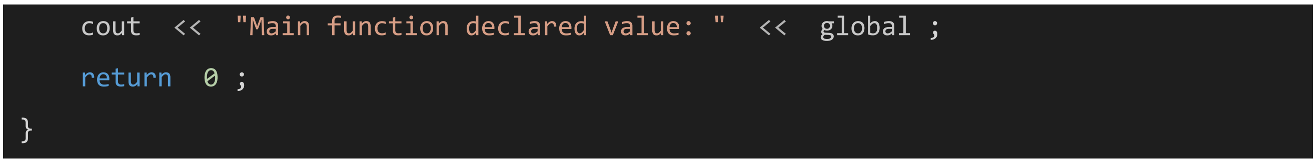
Practical 2.2 : Create a class called employee with the following details as variables within it.

```
#include< iostream>
using  namespace  std ;
class  Employee
{
    char  name [ 30] ;
    int  age ;
    char  des [ 10] ;
    long  salary ;
    public:
    void  Set() ;
    void  Put() ;
} ;
void  Employee  :: Set (){

    cout  <<  "Enter Employee Name : "  <<  endl ;
    cin  >>  name ;
    cout  <<  "Enter Employee Age : "  <<  endl ;
    cin  >>  age ;
    cout  <<  "Enter Employee Designation:"  <<  endl ;
    cin  >>  des ;
    cout  <<  "Enter Employee Salary : "  <<  endl ;
    cin  >>  salary ;
}
void  Employee  :: Put ()
{
    cout  <<  "Employee Name : "  <<  endl
```

```
          <<  name ;
    cout  <<  "Employee Age : "  <<  endl
          <<  age ;
    cout  <<  "Employee Designation:"  <<  endl
          <<  des ;
    cout  <<  "Employee Salary : "  <<  endl
          <<  salary ;
}
int  main ()
{
    Employee  A [ 3] ;
    int  i ;
    for  ( i  =  0 ;  i  <  3 ;  i ++ )
    {

        A [ i ]. Set() ;
    }
    for  ( i  =  0 ;  i  <  5 ;  i ++ )
    {

        A [ i ]. Put() ;
    }
    return  0 ;
}

```

Practical 2.3: WAP to illustrate the use of scope resolution operator. Display the various values of the same variables declared at different scope levels.

```
#include < iostream>
using  namespace  std ;
int  global  =  5 ;
int  main ()
{
    int  global  =  10 ;
    cout  <<  "Global value: "  <<  :: global  <<  endl ;
```



```
    cout  <<  "Main function declared value: "  <<  global ;
    return  0 ;
}
```

4)ERRORS ENCOUNTERED DURING PROGRAM'S EXECUTION

(Kindly jot down the compile time errors encountered)

None

5)PROGRAMS' EXPLANATION (in brief)

Practical 2.1 : 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.

In this program, we find the address using the formula "B + W * ((I – LR) * N + (J – LC)) " where

I = Row Subset of an element whose address to be found,
J = Column Subset of an element whose address to be found, B = Base address,
W = Storage size of one element store in any array(in bytes),
LR = Start row index of matrix(in this case we consider it as zero),
LC = Start column index of matrix(in this case we consider it as zero), M = Number of rows given in the matrix.

Practical 2.2: Create a class called employee with the following details as variables within it.

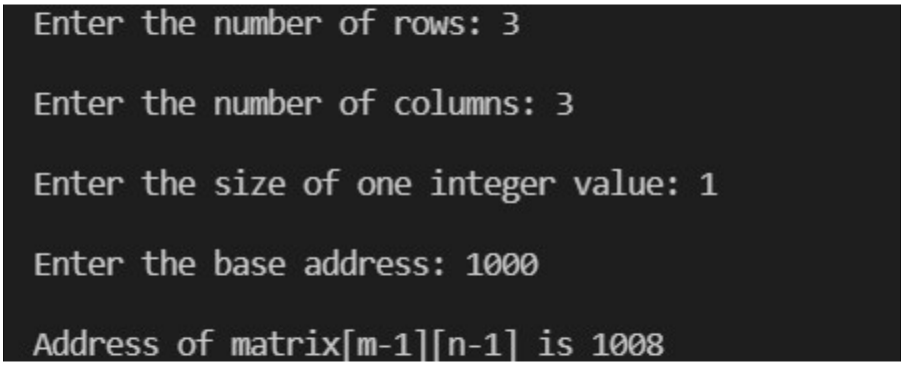
In this program, we create a class and put the required values and functions in it. Then we simply use the function inside the loop to take input and show the details of the employee.

Practical 2.3: WAP to illustrate the use of scope resolution operator. Display the various values of the same variables declared at different scope levels.

In this program, we used declare variable with different scope levels and access global value using "::" inside main function and display the desired output.

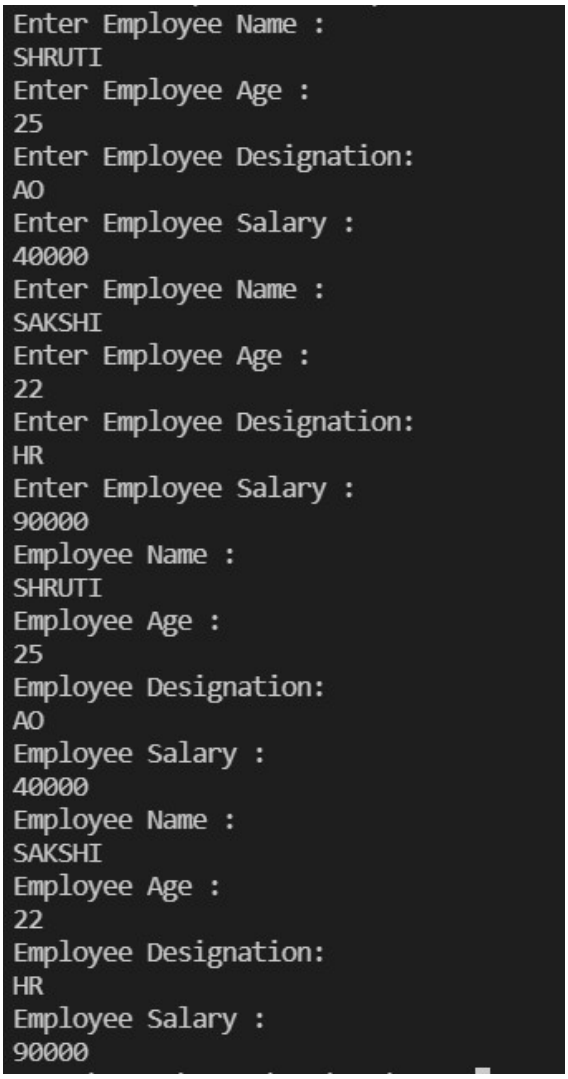
5)OUTPUT

PROGRAM 2.1



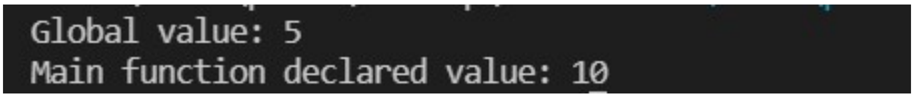
```
Enter the number of rows: 3

Enter the number of columns: 3

Enter the size of one integer value: 1

Enter the base address: 1000

Address of matrix[m-1][n-1] is 1008
```

PROGRAM 2.2



```
Enter Employee Name :
SHRUTI
Enter Employee Age :
25
Enter Employee Designation:
AO
Enter Employee Salary :
40000
Enter Employee Name :
SAKSHI
Enter Employee Age :
22
Enter Employee Designation:
HR
Enter Employee Salary :
90000
Employee Name :
SHRUTI
Employee Age :
25
Employee Designation:
AO
Employee Salary :
40000
Employee Name :
SAKSHI
Employee Age :
22
Employee Designation:
HR
Employee Salary :
90000
```

PROGRAM 2.3



```
Global value: 5
Main function declared value: 10
```

LEARNING OUTCOMES

- Understand the concepts of object-oriented programming including programming process and compilation process.
- Apply different techniques to decompose a problem and programmed a solution with its sub modules.
- Analyze and explain the behavior of simple programs involving the programming addressed in the course.
- Implement and evaluate the programs using the syntax and semantics of object-oriented programming.
- Design the solution of real-world problems in order to determine that the program performs as expected .

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	