CS201-Lab-Exercise - this is the latest lab from cs201, you can practice it and also another cs students

Introduction to programming /c++ (Virtual University of Pakistan)

Lab # 13

Week = 16 August – 20 August-2021

Problem Statement:

You need to write a program in which the following concepts must be implemented.

Write a template function name “Add” which accepts two arguments of the same type int, float, double from the user then add those arguments, return their sum and display them on the screen.

Write a class named as firstClass, also write another class named as secondClass which should be nested inside the firstClass. Define a function named displayMessage inside the secondClass, this method should print the message “Inside the second class” on the screen. In the main function create the object of the secondClass and invoke the displayMessage method using the secondClass object.

Solution:

```
#include <iostream>

using namespace std;


class firstClass {


public:


	class secondClass {

		public:

			void displayMessage(){

			cout<< "Function inside the inner class" <<endl;

			}

	};

};


template<class T>
```

```
T Add (T x, T y) {
return x + y;
}


int main() {

firstClass :: secondClass object;
object.displayMessage();

int integerOne = 3, integerTwo = 5;
float floatOne = 12.34, floatTwo = 894.4;
double doubleOne = 1236.58, doubleTwo = 8945.685;

cout<< "Addition of Two integer Number is = " << Add(integerOne,integerTwo) <<endl;
cout<< "Addition of Two floating point Number is = " << Add(floatOne,floatTwo) <<endl;
cout<< "Addition of Two double Number is = " << Add(doubleOne , doubleTwo) <<endl;

return 0;
}
```

Lab # 12

Week = 9 August – 13 August-2021

Problem Statement:

Write a program in C++ that add two class objects by operator overloading “plus (+)” operator. You are required to create a class named “MathClass“ and declare the class member and member functions. Also declare a class data member named as “number” and a parameterized constructor which take one argument and initializes the number. Define the ‘+’ operator overloaded function to add two object’s numbers. Also define another member function named as “Display ()” that shows the calculation result. Create three class objects for example: obj1, obj2 and result. Values are passed by calling the parameterized constructor in main () function. Add two object values by calling the ‘+’ operator overloaded function and then call display () function using obj1, obj2 and result.

Solution:

```
#include <iostream>

using namespace std;

class MathClass

{

private:

int number;

public:

MathClass()

    {

number = 0 ;

    }


MathClass(int x)

    {

number = x ;

    }


MathClass operator +(MathClass m) {
```

```
MathClass temp;
temp.number= number+m.number;
return temp;


    }
void Display() { cout<<"Result: "<<number<<endl; }
};
int main()
{
MathClass first(4), second(2), result;
result = first + second;
result.Display();
system("pause");
}
```

Lab # 11

Week = 2 August - 6 August 2021

Problem Statement:

Write a C++ program in which you have to implement a class "**Rectangle**". The class will have the following data members:

- Double length
- Double breadth
- Double area

You have to implement a parameterized constructor of this class which takes the two parameters "length" and "breadth" and then calculate the area of the rectangle. Write getter and setter methods or member functions of the class. You have to write a **printArea()** function which will display the area of the rectangle. Also, you have to write a friend function of the class named as **friendofRect().**

In main () function you have to display the area by the following ways:

1. First create object with parameterized constructor and then print the Area of the rectangle
2. Secondly create pointer to object with new keyword and parameterized constructor and then call print the Area of the rectangle.
3. Update the values of length and breadth of first object using friend function and then print the Area of the rectangle.

Solution:

```
#include <iostream>
#include <stdlib.h>
using namespace std;

class Rectangle{
        private:
                double area;
                double length;
                double breadth

        public:
;
                //Parameterized Constructor of the Class
                Rectangle(double,double);

                //Setter and Getter Functions
                void setLength(double);
```

```
                void setBreadth(double);
                double getLength();
                double getBreadth();

                //Print Member function
                void printArea();

                //Friend function of Class
                friend void friendOfRect(Rectangle &);
};

//Definition of Parameterized Constructor
Rectangle::Rectangle (double length, double breadth){
        this->area= length*breadth;
}

//Definition of Setter and Getter Functions
void Rectangle::setLength(double length){
        this->length = length;
}
void Rectangle::setBreadth(double breadth){
        this->breadth = breadth;
}

double Rectangle::getLength(){
        return length;
}

double Rectangle::getBreadth(){
        return breadth;
}

//Definition of Print Member Function
void Rectangle::printArea(){
        cout<< "Area of the Rectangle is: " << area <<endl;
}

//Definition of Friend Function
void friendOfRect(Rectangle &r){
        r.length = 5.0;
        r.breadth = 3.0;
        r.area = r.length * r.breadth;
}
```

```
int main(){

	Rectangle r1(7,3);
	r1.printArea();

	Rectangle *r2 = new Rectangle(4,6);
	r2->printArea();

friendOfRect(r1);
	r1.printArea();

	system("pause");

                                                }
```

Lab # 10

Week = 26 July – 30 July-2021

Problem Statement:

Write a C++ program in which implement a class named "**Employee**". This class has the following data members:

- Char string name
- Double id
- Character string gender
- Integer age

You have to implement the default and a parameterized constructor for this class.

Write getters and setters for each data member of the class and also implement a member function named "**display**" that will output the values of these data members for the calling object.

In the main () function, you need to create two objects of class "Employee". Initialize one object with default constructor and other with parameterized constructor. Also show the default values of both objects with the display function.

In the end, update the values of both objects using setter functions and then show the updated values of data members of both objects using getter functions.

Solution:

```
#include <iostream>

//#include <string.h>

#include <cstring>

using namespace std;



class Employee {


    private:

        char name[30];

        char gender[10];

        int age;
```

```
        double id;

    public:

        Employee();
        Employee(char[], char[], int, double);

        void setName(char[]);
        void setGender(char[]);
        void setAge(int);
        void setId(double);

        char* getName();
        char* getGender();
        int getAge();
        double getId();

        void display();

};

Employee::Employee()
 {
    strcpy(name, "Empty");
    strcpy(gender, "Empty");
    age = 0;
    id = 0.0;
```

```
}

Employee::Employee(char Name[], char Gender[], int Age, double Id)
{
        strcpy(name, Name);
        strcpy(gender, Gender);
        age = Age;
        id = Id;
}

void Employee::setName(char Name[])
 {
        strcpy (name, Name);

 }

void Employee::setGender(char Gender[])
 {
        strcpy (gender, Gender);

 }

void Employee::setAge(int Age)
  {
        age = Age;

  }
```

```
void Employee::setId(double Id)
  {
        id = Id;

  }


char* Employee::getName()
   {
        return name;

        }

        char* Employee::getGender()
   {
        return gender;

        }

        int Employee::getAge()
         {
              return age;

         }

        double Employee::getId()
          {
              return id;
```

```
}

void Employee::display()
{
    cout<<endl<< "The name of the Employee is " << name <<endl;
    cout<< "The gender of the Employee is " << gender <<endl;
    cout<< "The age of the Employee is " << age <<endl;
    cout<< "The Id of the Employee is " << id <<endl;

}

int main () {

   Employee e1;
   Employee e2("Amir", "Male", 35, 166);

    e1.display();
e2.display();

e1.setName("Asif");
e1.setGender("Male");
e1.setAge(30);
e1.setId(162);

cout<<endl<< "The name of the Employee is " << e1.getName() <<endl;
```

```
        cout<< "The gender of the Employee is " << e1.getGender() <<endl;
        cout<< "The age of the Employee is " << e1.getAge() <<endl;
        cout<< "The Id of the Employee is " << e1.getId() <<endl;


system("pause");
return 0;

}
//ends here
```

Lab # 9

Week = 12 July – 16 July-2021

Problem Statement:

Write a program that overloads function named “myfunc” for integer, double and character data types. For example, if an integer value is passed to this function, the message "**using integer myfunc"** should be printed on screen, on passing a double type value, the message "**using double myfunc"** and on passing character value, the message "**using character myfunc"** should get printed.

Solution:

```
#include <iostream>

using namespace std;


int myfunc (int x);

double myfunc (double y);

char myfunc(char z);

main()

{

cout<<myfunc(5) << "\n";

cout<<myfunc(5.5) << "\n";

cout<<myfunc('x') << "\n";


system("pause");

}


Int myfunc(int x)

{

cout<< "Using integer myfunc()\n";

return x;
```

```
}
Double myfunc(double y)
{
cout<< "Using float myfunc()\n";
return y;
}
Char myfunc(char z)
{
cout<< "Using char myfunc()\n";
return z;
}
```

Lab # 8

Week = 5 July – 9 July-2021

Problem Statement:

Write a program which declare two variables of integer type and take their values as input from user. Also, perform the following operations on them and print their values.

1. Bitwise and Logical AND
2. Bitwise and Logical OR

Solution:

```
#include <iostream>

using namespace std;

int main(){

    int var1 = 0, var2 = 0, var3 = 0;

    cout<< "Enter First Value: ";

    cin>> var1;


    cout<< "Enter Second Value: ";

    cin>> var2;


    var3 = var1 && var2;

    cout<<"Logical AND is "<< var3 <<endl;

    var3 = var1 & var2;

    cout<<"Bitwise AND is "<< var3<<endl;

        var3 = var1 || var2;

    cout<<"Logical OR is "<< var3 <<endl;

    var3 = var1 | var2;

    cout<<"Bitwise OR is "<< var3<<endl;


}
```

Lab # 7

Week = 14 June – 18 June-2021

Problem Statement:

Write a program which has:

1. A structure with only two member variables of integer and float type.
2. Initialize two data members by assigning 0 values in different ways.
3. Take input from user to assign different values to both structure variables.
4. Write a function that takes two structure variables as parameters.
5. This function must return a variable of structure type.
6. Function must add the data members of two passed structures variables and store the values in a new structure variable and print it on the screen.

Solution:

```
//Adding structure variable values using function

#include <iostream>

using namespace std;


//Structure with two Data Members

struct MyStruct{

    int i;

    float f;

}

ms3 = {0,0.0}, // To show different methods of initializing structure

ms4 = {0,0.0};


//Function to add two structure variables

MyStruct add(MyStruct ms1, MyStruct ms2){

     MyStruct ms3;

     ms3.i = ms1.i + ms2.i;

     ms3.f = ms1.f + ms2.f;

     return ms3;

}
```

```
main(){
    //Initialize structure variable values
    MyStruct ms1 = {0,0.0};
    struct MyStruct ms2 = {0,0.0};

    cout <<"Enter integer value of First structure variable ";
    cin>> ms1.i;
    cout <<"Enter float value of First structure variable ";
    cin>> ms1.f;

    cout <<"\n Enter integer value of Second structure variable ";
    cin>> ms2.i;
    cout <<"Enter float value of Second structure variable ";
    cin>> ms2.f;

    cout<< "\n values of data members of first structure variable\n\n";
    cout << ms1.i << "\t" << ms1.f << endl;

    cout<< "values of data members of Second structure variable\n\n";
    cout << ms2.i << "\t" << ms2.f <<endl;

    MyStruct ms3= add(ms1,ms2);
    cout<< "values of data members of structure variable returned from function\n\n";
    cout << ms3.i << "\t" << ms3.f <<endl;

    system("pause");
                                                    }
```

Lab # 6

Week = 7 June – 11 June-2021

Problem Statement:

Write a program in which you need to declare an integer type matrix of size 4*4. In this program:

1. You should take input values from the users and store it in 4*4 matrix.
2. Display this matrix on the screen.
3. Also, Display the transpose of this matrix by converting rows into cols.

Note: Use different functions for above three point's functionality.

Solution:

```
#include <iostream>

using namespace std;

const int arraySize = 4;

void readMatrix(int arr[][arraySize]);

void displayMatrix(int a[][arraySize]);


main(){
	int a[arraySize][arraySize];


	// Taking input from user

	readMatrix(a);


	// Display the matrix

	cout << "\n\n" << "The original matrix is: " << '\n';

	displayMatrix(a);


	system("pause");

}


void readMatrix(int arr[arraySize][arraySize]){
```

```
	int row, col;
	for (row = 0; row < arraySize; row ++){
		for(col=0; col < arraySize; col ++){
			cout << "\n" << "Enter " << row << ", " << col << " element: ";
			cin >> arr[row][col];
		}
	cout << '\n';
	}
}

void displayMatrix(int a[][arraySize]){
	int row, col;
	for (row = 0; row < arraySize; row ++){
		for(col = 0; col < arraySize; col ++){
			cout << a[row][col] << '\t';
		}
	cout << '\n';
	}
}
//end of lab=6
```

Lab # 5

Week = 31 May – 4 June-2021

Problem Statement:

Write a program that will take

1. Two strings names as input from the user in the form of character arrays namely as “firstArray” and “secondArray” respectively.
2. Both arrays along with the size will be passed to the function “CompareStrings”.
3. CompareStrings function will use pointer to receive arrays in function and then start comparing both arrays by using while loop.
4. If all the characters of both these arrays are same then the message “Both strings are same” should be displayed on the screen.

Note: For comparing both these arrays, the size should be same.

Solution:

```
#include <iostream>

#include <cstring>

using namespace std;


// function definition

void CompareStrings(char *arr1, char *arr2, int size){


        int check = 1;

        int i = 0;


        while(i<size){

if(arr1[i] != arr2[i]){

                        check = 0;

                        break;

                }


i++;
```

```
    }


        if(check == 1)
        cout<< "Both strings are same" <<endl;
        else
        cout<< "Both strings not same " <<endl;
}
int main() {
        char firstArray[20], secondArray[20];
        cout<< "Enter the first Name: ";
        cin.getline(firstArray, sizeof(firstArray));

        cout<< "Enter the second Name: ";
        cin.getline(secondArray, sizeof(secondArray));



        if(strlen(firstArray) == strlen(secondArray)){
                int size = strlen(firstArray);
                CompareStrings (firstArray, secondArray, size);
        }
        else {
                cout<< "Size of both names are not same" <<endl;
        }


        system("pause");
                                                }// Lab 5 ends here
```

Lab # 4

Week = 24 May – 28 May-2021

Problem Statement:

Write a program in which you have to declare an integer array of size 10 and initializes it with numbers of your choice. Find the maximum and minimum number from the array and output the numbers on the screen.

For finding the maximum and minimum numbers from the array you need to declare two functions **findMax** and **findMin** which accept an array and size of array (an int variable) as arguments and find the max min numbers, and return those values.

Solution:

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


int findMin(int [],int);

int findMax(int [],int);

int main() {

const int SIZE = 10;

int number[10] = {

        21,25,89,83,67,81,52,100,147,10
};


cout<< "Maximum number in the array is :" <<findMax(number, SIZE) <<endl;

cout<< "Minimum number in the array is :" <<findMin(number, SIZE) <<endl;


return 0;

}

int findMin(int array[],int size){

        int min = 0;

        min = array[0];
```

```
	for (int i = 0; i<size; i++){
		if(min > array[i])
			min = array[i];
	}
return min;
}


int findMax(int array[],int size){
	int max = 0;

	max = array[0];
	for (int i = 0; i<size; i++){
		if(max < array[i])
		max = array[i];
	}
	return max;
}
```

Lab # 3

Week = 18 May – 21 May-2021

Problem Statement:

Write a program in which you have to define a function displayDiagnol which will have two integer arguments named rows and cols. In the main function, take the values of rows and columns from the users. If the number of rows is same as numbers of columns then call the displayDiagnol function else show a message on screen that number of rows and columns is not same.

The following logic will be implemented inside the displayDiagnol function:

The function will take the value of rows and cols which are passed as argument and print the output in matrix form. To print the values in the matrix form, nested loops should be used. For each loop, you have to use a counter variable as counter. When the value of counters for each loop equals, then it prints the value of row at that location and prints hard coded zero at any other location.

Example if the user enters rows and cols as 3, then the output should be like this

1 0 0

0 2 0

0 0 3

Example: when rows and columns are not same.

```
Enter the number of rows:3
Enter the number of columns:2
Wrong input! Num of rows should be equal to num of columns
--------------------------------
```

Example: when rows and columns are same.

```
Enter the number of rows:3
Enter the number of columns:3
1 0 0
0 2 0
0 0 3

--------------------------------
```

Solution:

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

void displayDiagonal(int,int); // function declaration
```

```
int main(){
	int rows, columns;
	rows = 0;
	columns = 0;

	cout<< "Enter the number of rows:";
	cin>> rows;

	cout<< "Enter the number of columns:";
	cin>> columns;

	if(rows == columns)
	displayDiagonal(rows,columns); // call function
	else
	cout<< "Wrong input! Num of rows should be equal to num of columns";

	return 0;
}

// function definition
void displayDiagonal(int rows, int columns){

	for (int i = 1; i<=rows; i++) {
		for (int j = 1; j<=columns; j++){

			if(i==j)
			cout<<i<< " ";
```

```
                    else
                    cout<< 0 << " ";
            }
            cout<< "\n";
        }
}
```

Lab # 2

Week = 3 May – 7 May-2021

Problem Statement: While Loop

"Calculate the average age of a class of ten students using while loop. Prompt the user to enter the age of each student."

- We need 10 values of variable age of int type to calculate the average age.
  int age;

- "Prompt the user to enter the age of each student" this requires cin>> statement.
  For example:
  cin>> age;

- Average can be calculated by doing addition of 10 values and dividing sum with 10.

  TotalAge = TotalAge + age1;

  AverageAge = TotalAge / 10;

Solution:

```
#include<iostream>

using namespace std;


main() {


    int age=0;

    int TotalAge = 0, AverageAge = 0;

    int ageCounter=0;

    while (ageCounter <10)

    {

        cout<<"please enter the age of student "<<++ageCounter<<" :\t";

        cin>>age;
```

```
            TotalAge = TotalAge + age;
        }

         cout<< "Total Age of 10 students = "TotalAge<<endl;
        AverageAge = TotalAge/10;

        //Display the result (average age)
        cout<<"Average of students is: "<<AverageAge;
}
```

Lab # 1

Week = 26-April -- 30-April-2021

Problem Statement:

"Calculate the average age of a class of ten students. Prompt the user to enter the age of each student."

- We need 10 variables of int type to calculate the average age.
  int age1, age2, age3, age4, age5, age6, age7, age8, age9, age10;

- "Prompt the user to enter the age of each student" this requires cin>> statement.
  For example:
  cin>> age1;

- Average can be calculated by doing addition of 10 variables and dividing sum with 10.

  TotalAge = age1 + age2 + age3 + age4 + age5 + age6 + age7 + age8 +age9 + age10 ;

  AverageAge = TotalAge / 10;

Solution:

```
#include<iostream>

using namespace std;


main() {


    int age=0;

    int TotalAge = 0, AverageAge = 0;


    for(int i = 1;i<=10;i++)

    {

        cout<<"please enter the age of student "<<i<<" :\t";

        cin>>age;
```

```
            TotalAge += age;
        }


        AverageAge = TotalAge/10;


        //Display the result (average age)
        cout<<"Average of students is: "<<AverageAge;
}
```