*Object Oriented Programming Methodology*

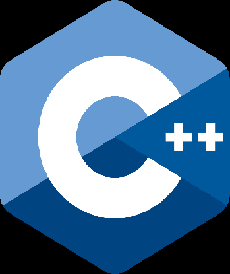
*III Semester*

*Electronics and Computer Science Engineering*

*Prepared by,*

*Prof. Archana Lopes / Prof. Prachi Patil*





|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Rutik Yerunkar | | |
| **Lab Experiment No.** | 1 | **Roll No.** | 8851 |
| **Date Of Performance.:** | 01-10-20 | **Date Of Sub.:** | 01-10-20 |
| **Expt. Title** | Read two numbers and apply all types of operators printing their results | | |
| **CO Mapping** | ECL305.1 | | |

**Problem Definition:**

Write a C++ program to read two numbers and apply all types of operators printing their results

**Objective of the Experiment:** Understanding the use of various operators and input statements in C++

|  |
| --- |
| **Attach the Source code and output:** |

Source Code:

//Program to read two numbers and apply all types of operators.

#include<iostream>

using namespace std;

int main()

{

    int n1,n2;

    cout<<"Enter Two Number's:";

    cin>>n1>>n2;

    cout<<"Sum="<<n1+n2<<endl;

    cout<<"Difference="<<n1-n2<<endl;

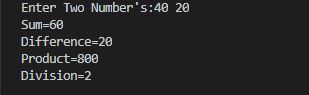
    cout<<"Product="<<n1\*n2<<endl;

    cout<<"Division="<<n1/n2<<endl;

    return 0;

}

Output:



|  |
| --- |
| The program was tested for different sets of inputs.  Program is working is SATISFACTORY NOT SATISFACTORY ( Tick appropriate outcome) |

**Post Lab Questions:**

1. WAP to print size of each datatype (basic + long,short,signed,….)

Source Code:

//To print size of each datatype

#include<iostream>

using namespace std;

int main()

{

    cout<<"Size Of Int Datatype:"<<sizeof(int)<<endl;

    cout<<"Size Of Float Datatype:"<<sizeof(float)<<endl;

    cout<<"Size Of Double Datatype:"<<sizeof(double)<<endl;

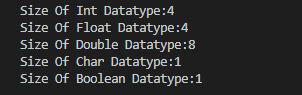
    cout<<"Size Of Char Datatype:"<<sizeof(char)<<endl;

    cout<<"Size Of Boolean Datatype:"<<sizeof(bool)<<endl;

    return 0;

}

Output:



**Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On time Completion and Submission (2)** | **Knowledge of the topic (4)** | **Implementation and Output (4)** | **Total (10)** |
|  |  |  |  |

**Date & Signature of teacher:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Rutik Yerunkar | | |
| **Lab Experiment No.** | 2 | **Roll No.** | 8851 |
| **Date Of Performance.:** | 01-10-20 | **Date Of Sub.:** | 01-10-20 |
| **Expt. Title** | Write a program that reads in two integers and determines and prints if the first is a multiple of a second. | | |
| **CO Mapping** | ECL305.2 | | |

**Problem Definition:**

Write a C++ program that reads in two integers and determines and prints if the first is a multiple of a second

**Objective of the Experiment:** Understanding the use of decision making statement (if statement)

|  |
| --- |
| **Attach the Source code and output:** |

Source Code:

//Program to read two numbers and determines and prints if first is multiple of second.

#include<iostream>

using namespace std;

int main()

{

    int n1,n2;

    cout<<"Enter Two Number's:";

    cin>>n1>>n2;

    if(n1%n2==0)

        cout<<"First Is Multiple Of Second.\n";

    else

        cout<<"First Is Not Multiple Of Second.\n";

    return 0;

}

Output:

Lab_Manual_Exp2_Output1.JPG

Lab_Manual_Exp2_Output2.JPG

|  |
| --- |
| The program was tested for different sets of inputs.  Program is working is SATISFACTORY NOT SATISFACTORY ( Tick appropriate outcome) |

**Post Lab Questions:**

1. Write a C++ program that asks the user to enter two integers, obtains the numbers from the user, then prints the larger number followed by the words " is larger". If the numbers are equal, print the message "These numbers are equal"

2. A set of two linear equations with two unknowns x1 and x2 is given below:

ax1+bx2=m

cx1+dx2=n

the set has a unique solution

x1=(md-bn)/(ad-cd)

x2=(na-mc)/(ad-cd)

provided the denominator ad-cd is not equal to 0. Write a C++ program that will read the values of constants a,b,c,d,m,n and compute x1 and x2. An appropriate meessage should be printed if ad-cb=0

Source Code For Q1. :

//1.To compare Two numbers

#include<iostream>

using namespace std;

int main()

{

    int n1,n2,large;

    cout<<"Enter Two Number's:";

    cin>>n1>>n2;

    if(n1==n2)

        cout<<"These Number's Are Equal.";

    else

    {

        large=n1>n2?n1:n2;

        cout<<large<<" Is Larger.";

    }

    return 0;

}

Output For Q1. :

Post_Lab_Exp2_Q1.Output_Part1.JPG

Post_Lab_Exp2_Q1.Output_Part2.JPG

Post_Lab_Exp2_Q1.Output_Part3.JPG

Source Code For Q2. :

//2.To find Roots Of Equation

#include<iostream>

using namespace std;

int main()

{

    int a,b,c,d,m,n;

    float x1,x2;

    cout<<"Enter Value Of a,b,c,d,m,n:";

    cin>>a>>b>>c>>d>>m>>n;

    if(a\*d-c\*d==0)

        cout<<"Enter Appropriate Value Of a,d and c\n";

    else

    {

        x1=(float)(m\*d-b\*n)/(a\*d-c\*d);

        x2=(float)(n\*a-m\*c)/(a\*d-c\*d);

        cout<<"X1="<<x1<<endl;

        cout<<"x2="<<x2<<endl;

    }

    return 0;

}

Output For Q2. :

Post_Lab_Exp2_Q2.Output_Part1.JPG

Post_Lab_Exp2_Q2.Output_Part2.JPG

**Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On time Completion and Submission (2)** | **Knowledge of the topic (4)** | **Implementation and Output (4)** | **Total (10)** |
|  |  |  |  |

**Date & Signature of teacher:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Rutik Yerunkar | | |
| **Lab Experiment No.** | 3 | **Roll No.** | 8851 |
| **Date Of Perf.:** | 06-10-20 | **Date Of Sub.:** | 11-10-20 |
| **Expt. Title** | Study decision making statement- switch statement | | |
| **CO Mapping** | ECL303.2 | | |

**Problem Definition:** Write a C++ program using nested switch to print the elective subject offered to students

|  |  |  |
| --- | --- | --- |
| Year | Branch | Subject name |
| 3 | IT | Artificial Intelliegence |
| 4 | IT | Storage Management |
| 3 | ECS | Green IT |
| 4 | ECS | Project Management |

**Objective of the Experiment:** Understanding the use of decision making statement (switch statement)

|  |
| --- |
| **Attach the Source code and output:** |

Source Code:

//Use switch statement to show elective subject

#include<iostream>

using namespace std;

int main()

{

    int year,x;

    cout<<"Enter The Number Of Year:";

    cin>>year;

    cout<<"1.IT\n2.ECS\nEnter Your Choice:";

    cin>>x;

    switch(year)

    {

        case 3: switch(x)

                {

                    case 1: cout<<"Elective Subject Name:Artificial Intelligence\n";

                            break;

                    case 2: cout<<"Elective Subject Name:Green IT\n";

                            break;

                }

                break;

        case 4: switch(x)

                {

                    case 1: cout<<"Elective Subject Name:Storage Management\n";

                            break;

                    case 2: cout<<"Elective Subject Name:Project Management\n";

                            break;

                }

                break;

        default: cout<<"Invalid Choice\n";

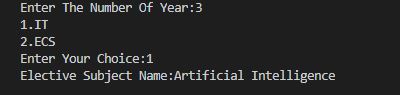
                break;

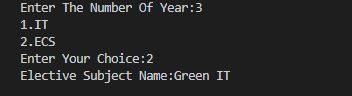
    }

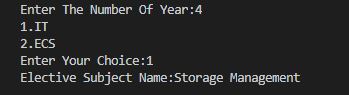
    return 0;

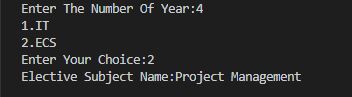
}

Output:









|  |
| --- |
| The program was tested for different sets of inputs.  Program is working is SATISFACTORY NOT SATISFACTORY ( Tick appropriate outcome) |

**Post Lab Questions:**

1.A cloth showroom has announced following Discount on purchase. Find the net amount to be paid by the customer. Demonstrate using switch.

|  |  |  |
| --- | --- | --- |
| Purchase Amount | Mill cloth | Handloom cloth |
| 0-100 | - | 5% |
| 101-200 | 5% | 7.5% |
| 201-300 | 7.5% | 10% |
| Above 300 | 10% | 15% |
|  |  |  |

**Source Code:**

//A cloth showroom has announced following  Discount on purchase. Find the net amount to be paid by the customer.

#include<iostream>

using namespace std;

int main()

{

    int amt,choice,tot,discount;

    cout<<"Enter The Purchase Amount:";

    cin>>amt;

    cout<<"1.Mill Cloth\n2.Handloom Cloth\nEnter Your Choice:";

    cin>>choice;

    if(amt<=100)

    {

        switch (choice)

        {

            case 1: discount=0;

                    cout<<"Discount Unavailable\n";

                    break;

            case 2: discount=amt\*0.05;

                    cout<<"Discount Received:"<<discount<<endl;

                    break;

            default: cout<<"Invalid Choice\n";

                     break;

        }

    }

    else if (amt>=101 && amt<=200)

    {

        switch(choice)

        {

            case 1: discount=amt\*0.05;

                    cout<<"Discount Received:"<<discount<<endl;

                    break;

            case 2: discount=amt\*0.075;

                    cout<<"Discount Received:"<<discount<<endl;

                    break;

            default: cout<<"Invalid Choice\n";

                    break;

        }

    }

    else if (amt>=201 && amt<=300)

    {

        switch (choice)

        {

            case 1: discount=amt\*0.075;

                    cout<<"Discount Received:"<<discount<<endl;

                    break;

            case 2: discount=amt\*0.1;

                    cout<<"Discount Received:"<<discount<<endl;

                    break;

            default: cout<<"Invalid Choice\n";

                    break;

        }

    }

    else if (amt>300)

    {

        switch (choice)

        {

            case 1: discount=amt\*0.1;

                    cout<<"Discount Received:"<<discount<<endl;

                    break;

            case 2: discount=amt\*0.15;

                    cout<<"Discount Received:"<<discount<<endl;

                    break;

            default: cout<<"Invalid Choice\n";

                    break;

        }

    }

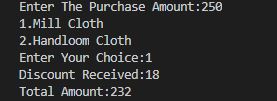
    tot=amt-discount;

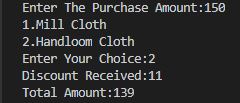
    cout<<"Total Amount:"<<tot;

    return 0;

}

**Output:**

****

****

**Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On time Completion and Submission (2)** | **Knowledge of the topic (4)** | **Implementation and Output (4)** | **Total (10)** |
|  |  |  |  |

**Date & Signature of teacher:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Rutik Yerunkar | | |
| **Lab Experiment No.** | 4 | **Roll No.** | 8851 |
| **Date Of Perf.:** | 06-10-20 | **Date Of Sub.:** | 11-10-20 |
| **Expt. Title** | Study of looping statements | | |
| **CO Mapping** | ECL305.2 | | |

**Problem Definition:** Write a program that calculates how much money you’ll end up with if you invest an amount of money at a fixed interest rate, compounded yearly. Have the user furnish the initial amount, the number of years, and the yearly interest rate in percent. Some interaction with the program might look like this:

Enter initial amount: 3000

Enter number of years: 10

Enter interest rate (percent per year): 5.5

At the end of 10 years, you will have 5124.43 dollars.

**Objective of the Experiment:** Understanding the use of looping statements

|  |
| --- |
| **Attach the Source code and output:** |

Source Code:

/\* Write a program that calculates how much money you’ll end up with if you invest an amount of money at a fixed interest rate, compounded

yearly. Have the user furnish the initial amount, the number of years, and the yearly interest rate in percent.\*/

#include<iostream>

using namespace std;

int main()

{

    int amt,yr,i;

    float roi,tot;

    cout<<"Enter The Initial Amount:";

    cin>>amt;

    cout<<"Enter number of years:";

    cin>>yr;

    cout<<"Enter interest rate (percent per year):";

    cin>>roi;

    roi=roi/100;

    tot=amt;

    for(i=0;i<yr;i++)

    {

        tot=tot+(tot\*roi);

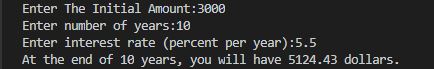
    }

    cout<<"At the end of "<<yr<<" years, you will have "<<tot<<" dollars.";

    return 0;

}

Output:



|  |
| --- |
| The program was tested for different sets of inputs.  Program is working is SATISFACTORY NOT SATISFACTORY ( Tick appropriate outcome) |

**Post Lab Questions:**

1. Write a C++ program to display Fibonacci series upto 'n' terms

Source Code:

//Write a C++ program to display Fibonacci series upto 'n' terms

#include<iostream>

using namespace std;

int main()

{

    int n1=0,n2=1,n,i,next;

    cout<<"Enter Number Of Terms:";

    cin>>n;

    cout<<"Fibonacci Series:\n"<<n1<<endl<<n2<<endl;

    for (i=1;i<=n-2;i++)

    {

        next=n1+n2;

        cout<<next<<endl;

        n1=n2;

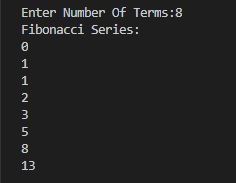
        n2=next;

    }

    return 0;

}

Output:



**Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On time Completion and Submission (2)** | **Knowledge of the topic (4)** | **Implementation and Output (4)** | **Total (10)** |
|  |  |  |  |

**Date & Signature of teacher:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Rutik Yerunkar | | |
| **Lab Experiment No.** | 5 | **Roll No.** | 8851 |
| **Date Of Perf.:** | 13-10-20 | **Date Of Sub.:** | 29-11-20 |
| **Expt. Title** | Study of arrays and strings | | |
| **CO Mapping** | ECL305.2 | | |

**Problem Definition:** Write a C++ function called reversit() that reverses a string (an array of char).The string should be passed to reversit() as an argument.Write a program to exercise reversit(). The program should get a string from the user, call reversit(), and print out the result. Use an input method that allows embedded blanks.

.

**Objective of the Experiment:** Understanding the use of strings

|  |
| --- |
| **Attach the Source code and output:** |

Source Code:

#include<iostream>

#include<cstring>

using **namespace** std;

**void** reverseit(**char** s[100],**int** l)

{

**char** temp;

    for(**int** i=0;i<=l/2-1;i++)

    {

        temp=s[i];

        s[i]=s[l-i-1];

        s[l-i-1]=temp;

    }

    cout<<"Reverse String Is:"<<s;

}

**int** main()

{

**char** s[100];

    cout<<"Enter String:";

    cin.get(s,100);

    reverseit(s,strlen(s));

    return 0;

}

Output:



|  |
| --- |
| The program was tested for different sets of inputs.  Program is working is SATISFACTORY NOT SATISFACTORY ( Tick appropriate outcome) |

**Post Lab Questions:**

1. Write a C++ program to search an element in an array using binary search

2. Write a C++ program to transpose a matrix.

Source Code For Q1.:

#include<iostream>

using **namespace** std;

**int** binarysearch(**int** a[],**int** k,**int** l,**int** u)

{

    if(l>u)

        return -1;

**int** mid;

    mid=(l+u)/2;

    if (k==a[mid])

        return 1;

    else if(k<a[mid])

        binarysearch(a,k,l,mid-1);

    else if(k>a[mid])

        binarysearch(a,k,mid+1,u);

}

**int** main()

{

**int** n,i,z,key;

    printf("Enter number of elements:");

    scanf("%d",&n);

**int** a[n];

    for(i=0;i<n;i++)

    {

        printf("Enter Data:");

        scanf("%d",&a[i]);

    }

    printf("Enter the Search Key:");

    scanf("%d",&key);

    z=binarysearch(a,key,0,n-1);

    if(z==1)

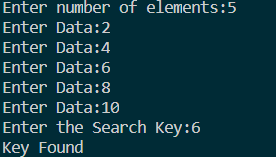
        printf("Key Found\n");

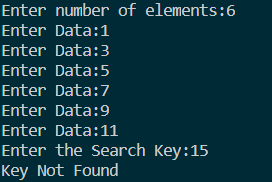
    else

        printf("Key Not Found\n");

}

Output For Q1.:





Source Code For Q2.:

#include<iostream>

using **namespace** std;

**void** accept(**int** x[10][10], **int** m, **int** n)

{

**int** i,j;

    for(i=0;i<=m-1;i++)

    {

        for(j=0;j<=n-1;j++)

        {

            cout<<"Enter Value:";

            cin>>x[i][j];

        }

    }

}

**void** display(**int** x[10][10],**int** m,**int** n)

{

**int** i,j;

    for(i=0;i<=m-1;i++)

    {

        for(j=0;j<=n-1;j++)

        {

            cout<<x[i][j]<<"\t";

        }

        cout<<"\n";

    }

}

**void** transpose(**int** a[10][10],**int** b[10][10],**int** m,**int** n)

{

**int** i,j;

    for(i=0;i<=m-1;i++)

    {

        for(j=0;j<=n-1;j++)

        {

            b[j][i]=a[i][j];

        }

    }

}

**int** main()

{

**int** a[10][10],m,n,b[10][10];

    cout<<"Enter Number of Rows and Columns:";

    cin>>m>>n;

    accept(a,m,n);

    transpose(a,b,m,n);

    cout<<"Original Matrix:\n";

    display(a,m,n);

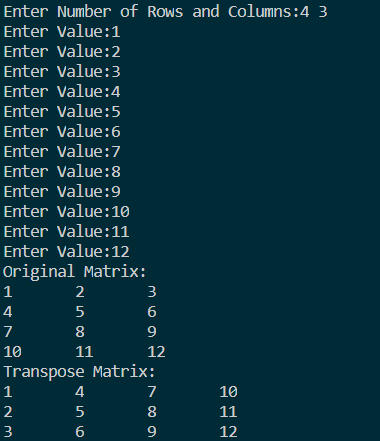
    cout<<"Transpose Matrix:\n";

    display(b,n,m);

    return 0;

}

Output For Q2.:



**Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On time Completion and Submission (2)** | **Knowledge of the topic (4)** | **Implementation and Output (4)** | **Total (10)** |
|  |  |  |  |

**Date & Signature of teacher:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Rutik Yerunkar | | |
| **Lab Experiment No.** | 6 | **Roll No.** | 8851 |
| **Date Of Perf.:** | 13-10-20 | **Date Of Sub.:** | 29-11-20 |
| **Expt. Title** | Study of functions and recursive functions | | |
| **CO Mapping** | ECL305.2 | | |

**Problem Definition:**

Write a recursive function to convert Decimal number to binary number.

.

**Objective of the Experiment:** Understanding the use of recursive functions

|  |
| --- |
| **Attach the Source code and output:** |

Source Code:

#include<iostream>

using **namespace** std;

**int** convert(**int** dec)

{

    if(dec==0)

        return 0;

    else

    {

        return (dec%2+10\*convert(dec/2));

    }

}

**int** main()

{

**int** dec,ans;

    cout<<"Enter Decimal Number:";

    cin>>dec;

    ans=convert(dec);

    cout<<"Binary Equivalent:"<<ans;

    return 0;

}

Output:



|  |
| --- |
| The program was tested for different sets of inputs.  Program is working is SATISFACTORY NOT SATISFACTORY ( Tick appropriate outcome) |

**Post Lab Questions:**

1. Write a C++ program to search an element in an array using binary search

2. Write a parameterized function to calculate GCD using Dijkstra's Algorithm

Source Code For Q1.:

#include<iostream>

using **namespace** std;

**int** binarysearch(**int** a[],**int** k,**int** l,**int** u)

{

    if(l>u)

        return -1;

**int** mid;

    mid=(l+u)/2;

    if (k==a[mid])

        return 1;

    else if(k<a[mid])

        binarysearch(a,k,l,mid-1);

    else if(k>a[mid])

        binarysearch(a,k,mid+1,u);

}

**int** main()

{

**int** n,i,z,key;

    printf("Enter number of elements:");

    scanf("%d",&n);

**int** a[n];

    for(i=0;i<n;i++)

    {

        printf("Enter Data:");

        scanf("%d",&a[i]);

    }

    printf("Enter the Search Key:");

    scanf("%d",&key);

    z=binarysearch(a,key,0,n-1);

    if(z==1)

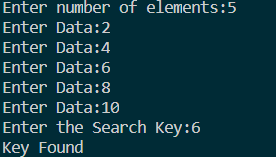
        printf("Key Found\n");

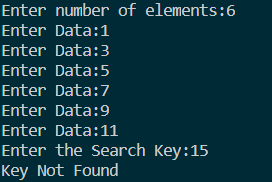
    else

        printf("Key Not Found\n");

}

Output For Q1.:





Source Code For Q2.:

#include<iostream>

using **namespace** std;

**int** gcd(**int** m, **int** n)

{

   if(m == n)

      return m;

   else if (m > n)

      return gcd(m-n, n);

   else

      return gcd(m, n-m);

}

**int** main()

{

**int** a,b,ans;

    cout<<"Enter two number's:";

    cin>>a>>b;

    ans=gcd(a,b);

    cout<<"GCD:"<<ans;

    return 0;

}

Output For Q2.:



**Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On time Completion and Submission (2)** | **Knowledge of the topic (4)** | **Implementation and Output (4)** | **Total (10)** |
|  |  |  |  |

**Date & Signature of teacher:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Rutik Yerunkar | | |
| **Lab Experiment No.** | 7 | **Roll No.** | 8851 |
| **Date Of Perf.:** | 20-10-20 | **Date Of Sub.:** | 29-11-20 |
| **Expt. Title** | Study of functions overloading | | |
| **CO Mapping** | ECL305.2 | | |

**Problem Definition:**

Write a overloaded function to swap two numbers.

.

**Objective of the Experiment:** Understanding the concept of polymorphism in OOPM.

|  |
| --- |
| **Attach the Source code and output:** |

Source Code:

#include<iostream>

using **namespace** std;

**void** swap(**int** a,**int** b)

{

**int** temp;

    temp=a;

    a=b;

    b=temp;

    cout<<"After Swapping n1="<<a<<" n2="<<b<<endl;

}

**void** swap(**float** a,**float** b)

{

**float** temp1;

    temp1=a;

    a=b;

    b=temp1;

    cout<<"After Swapping n3="<<a<<" n4="<<b<<endl;

}

**int** main()

{

**int** n1,n2;

**float** n3,n4;

    cout<<"Enter n1 and n2:";

    cin>>n1>>n2;

    cout<<"Enter n3 and n4:";

    cin>>n3>>n4;

    cout<<"Before Swapping n1="<<n1<<" n2="<<n2<<endl;

    cout<<"Before Swapping n3="<<n3<<" n4="<<n4<<endl;

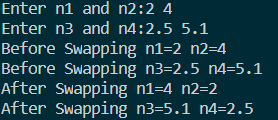
    swap(n1,n2);

    swap(n3,n4);

    return 0;

}

Output:



|  |
| --- |
| The program was tested for different sets of inputs.  Program is working is SATISFACTORY NOT SATISFACTORY ( Tick appropriate outcome) |

**Post Lab Questions:**

1. Write a C++ program including a inline function to find maximum of three numbers

Source Code:

#include<iostream>

using **namespace** std;

**inline** **int** largest(**int** **&**a,**int** **&**b,**int** **&**c)

{

**int** large=0;

    if(a>b)

        large=a;

    else

        large=b;

    if(c>large)

        large=c;

    return large;

}

**int** main()

{

**int** a,b,c,ans;

    cout<<"Enter Three Numbers:";

    cin>>a>>b>>c;

    ans=largest(a,b,c);

    cout<<"Largest Is:"<<ans;

    return 0;

}

Output:



**Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On time Completion and Submission (2)** | **Knowledge of the topic (4)** | **Implementation and Output (4)** | **Total (10)** |
|  |  |  |  |

**Date & Signature of teacher:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Rutik Yerunkar | | |
| **Lab Experiment No.** | 8 | **Roll No.** | 8851 |
| **Date Of Perf.:** | 27-10-20 | **Date Of Sub.:** | 29-11-20 |
| **Expt. Title** | Study of Structures | | |
| **CO Mapping** | ECL305.2 | | |

**Problem Definition:**

### Write a C++ program to write a structure student that accepts the name,cet marks and the physics, chemistry and maths marks. The program should display the the merit list by sorting it with the pcm (physics+chemistry+maths) marks and then by sorting it with the name of the student

.**Objective of the Experiment:** Understanding the concept of structures.

|  |
| --- |
| **Attach the Source code and output:** |

Source Code:

#include<iostream>

#include<cstring>

using **namespace** std;

**struct** student

{

**char** name[100];

**int** cet,phy,chem,math,total;

};

**void** sortingbymarks(student s[],**int** n)

{

**int** i,j;

    student temp;

    for(i=0;i<n-1;i++)

    {

        for(j=i+1;j<n;j++)

        {

            if(s[i].total<s[j].total)

            {

                temp=s[i];

                s[i]=s[j];

                s[j]=temp;

            }

        }

    }

}

**void** display(student s[],**int** n)

{

    cout<<"Name"<<"\t"<<"CET"<<"\t"<<"Phy"<<"\t"<<"Chem"<<"\t"<<"Maths"<<"\t""\t"<<"Total"<<endl;

    for(**int** i=0; i<=n-1;i++)

        cout<<s[i].name<<"\t"<<s[i].cet<<"\t"<<s[i].phy<<"\t"<<s[i].chem<<"\t"<<s[i].math<<"\t"<<s[i].total<<endl;

}

**void** sortingbyname(student s[],**int** n)

{

**int** i,j;

    student temp;

    for(i=0;i<n-1;i++)

    {

        for(j=i+1;j<n;j++)

        {

            if(strcmp(s[i].name,s[j].name)>0)

            {

                temp=s[i];

                s[i]=s[j];

                s[j]=temp;

            }

        }

    }

}

**int** main()

{

**int** n,i;

    student s[200];

    cout<<"Enter number of students:";

    cin>>n;

    for(i=0;i<=n-1;i++)

    {

        cout<<"Enter name:";

        cin>>s[i].name;

        cout<<"Enter marks in Physics:";

        cin>>s[i].phy;

        cout<<"Enter marks in Chemistry:";

        cin>>s[i].chem;

        cout<<"Enter marks in Maths:";

        cin>>s[i].math;

        cout<<"Enter marks in CET:";

        cin>>s[i].cet;

        s[i].total=s[i].phy+s[i].chem+s[i].math;

    }

    cout<<"Merit List:"<<endl;

    sortingbymarks(s,n);

    display(s,n);

    cout<<"Sorting according to name:"<<endl;

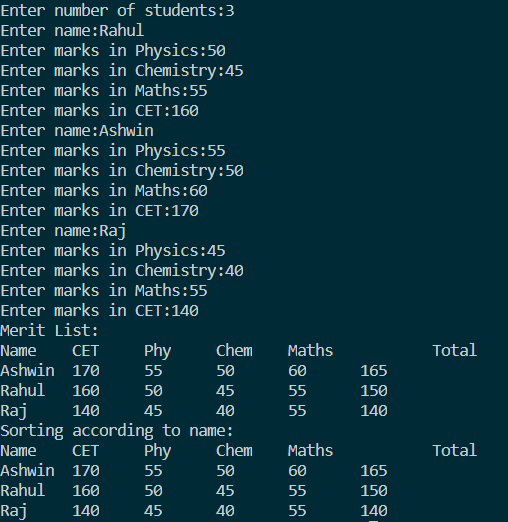
    sortingbyname(s,n);

    display(s,n);

    return 0;

}

Output:



|  |
| --- |
| The program was tested for different sets of inputs.  Program is working is SATISFACTORY NOT SATISFACTORY ( Tick appropriate outcome) |

**Post Lab Questions:**

1. Write a C++ Program to create a linked list using structures: add,delete and display node functions.

2. Create a structure 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 two time value in hours, minutes, and seconds. This can be in 12:59:59 format. Then converts each one to seconds (type int), adds these quantities, converts the result back to hours-minutes-seconds, stores the result in a time structure, and finally displays the result in 12:59:59 format.

Source Code For Q1.:

#include<iostream>

using **namespace** std;

**struct** node

{

**int** data;

**struct** node **\*** next;

};

**struct** node **\*** add(**struct** node **\*** head)

{

**int** n,i;

**struct** node **\*** temp;

    cout<<"Enter Number Of Nodes:";

    cin>>n;

    head=(**struct** node **\***) malloc(sizeof(struct node));

    cout<<"Enter Data:";

    cin>>head->data;

    temp=head;

    for(i=1;i<n;i++)

    {

        temp->next=(**struct** node **\***) malloc(sizeof(struct node));

        temp=temp->next;

        cout<<"Enter Data:";

        cin>>temp->data;

    }

    temp->next=NULL;

    return head;

}

**struct** node **\*** remove(**struct** node **\*** head)

{

**struct** node **\*** temp, \* temp1;

**int** pos,i=1;

    cout<<"Enter The Position:";

    cin>>pos;

    temp=head;

    while(i<pos)

    {

        temp1=temp;

        temp=temp->next;

        i++;

    }

    temp1->next=temp->next;

    temp->next=NULL;

    free(temp);

    return head;

}

**void** display(**struct** node **\*** head)

{

**struct** node **\***temp;

    temp=head;

    while(temp!=NULL)

    {

        cout<<temp->data<<" ";

        temp=temp->next;

    }

    cout<<"\n";

}

**int** main()

{

**int** choice;

**struct** node **\***head;

    head=NULL;

    do

    {

        cout<<"1.Add\n2.Display\n3.Delete\n4.Exit\nEnter Your Choice:";

        cin>>choice;

        switch(choice)

        {

            case 1: head=add(head);

                    break;

            case 2: display(head);

                    break;

            case 3:head=remove(head);

                    break;

            case 4: break;

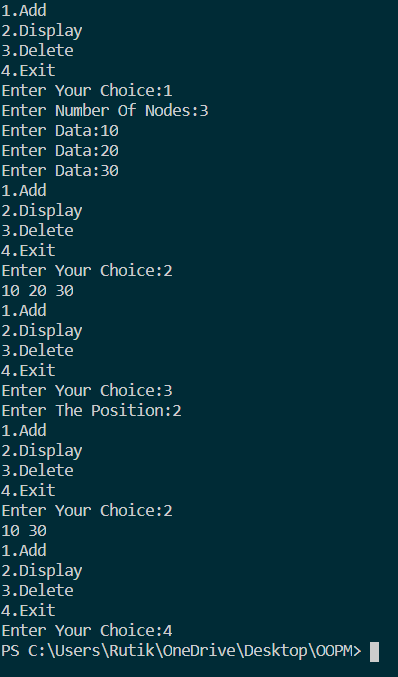
        }

    }while(choice!=4);

    return 0;

}

Output For Q1.:



Source Code For Q2.:

#include <iostream>

using **namespace** std;

**struct** time

{

**int** hours,mins,secs;

};

**void** addsecs(**struct** time **&**t, **struct** time **&**s)

{

    s.secs+= 60 \* 60 \* t.hours + 60 \* t.mins + t.secs;

}

**void** convert(**struct** time **&**s)

{

    s.hours = s.secs / (60 \* 60);

    s.mins = (s.secs - (60 \* 60 \* s.hours)) / 60;

    s.secs = (s.secs - (60 \* 60 \* s.hours) - (60 \* s.mins));

}

**int** main()

{

**struct** time t1,t2,s;

    cout << "TIME 1\nEnter Number of Hours:";

    cin >> t1.hours;

    cout << "Enter Number of Mins:";

    cin >> t1.mins;

    cout << "Enter Number of Seconds:";

    cin >> t1.secs;

    cout << "TIME 2\nEnter Number of Hours:";

    cin >> t2.hours;

    cout << "Enter Number of Mins:";

    cin >> t2.mins;

    cout << "Enter Number of Seconds:";

    cin >> t2.secs;

    addsecs(t1,s);

    addsecs(t2,s);

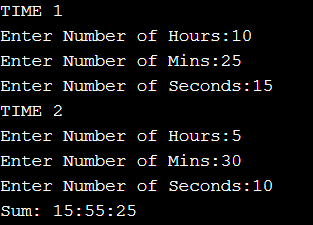
    convert(s);

    cout << "Sum: " << s.hours << ":" << s.mins << ":" << s.secs;

    return 0;

}

Output For Q2.:



**Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On time Completion and Submission (2)** | **Knowledge of the topic (4)** | **Implementation and Output (4)** | **Total (10)** |
|  |  |  |  |

**Date & Signature of teacher:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Rutik Yerunkar | | |
| **Lab Experiment No.** | 9 | **Roll No.** | 8851 |
| **Date Of Perf.:** | 27-10-20 | **Date Of Sub.:** | 29-11-20 |
| **Expt. Title** | Study of Classes and Objects | | |
| **CO Mapping** | ECL305.3 | | |

**Problem Definition:**

A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise “Required copies not in stock” is displayed. Design a system using a class called books with suitable member functions and constructors. Use new operator in constructors to allocate memory space required and also demonstrate destructors

.**Objective of the Experiment:** Understanding the concept of classes and objects.

|  |
| --- |
| **Attach the Source code and output:** |

|  |
| --- |
| The program was tested for different sets of inputs.  Program is working is SATISFACTORY NOT SATISFACTORY ( Tick appropriate outcome) |

**Post Lab Questions:**

1. Write a C++ Program to create a linked list using structures: add,delete and display node functions.

2. Create a structure 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 two time value in hours, minutes, and seconds. This can be in 12:59:59 format. Then converts each one to seconds (type int), adds these quantities, converts the result back to hours-minutes-seconds, stores the result in a time structure, and finally displays the result in 12:59:59 format.

Source Code For Q1.:

#include<iostream>

using **namespace** std;

**struct** node

{

**int** data;

**struct** node **\*** next;

};

**struct** node **\*** add(**struct** node **\*** head)

{

**int** n,i;

**struct** node **\*** temp;

    cout<<"Enter Number Of Nodes:";

    cin>>n;

    head=(**struct** node **\***) malloc(sizeof(struct node));

    cout<<"Enter Data:";

    cin>>head->data;

    temp=head;

    for(i=1;i<n;i++)

    {

        temp->next=(**struct** node **\***) malloc(sizeof(struct node));

        temp=temp->next;

        cout<<"Enter Data:";

        cin>>temp->data;

    }

    temp->next=NULL;

    return head;

}

**struct** node **\*** remove(**struct** node **\*** head)

{

**struct** node **\*** temp, \* temp1;

**int** pos,i=1;

    cout<<"Enter The Position:";

    cin>>pos;

    temp=head;

    while(i<pos)

    {

        temp1=temp;

        temp=temp->next;

        i++;

    }

    temp1->next=temp->next;

    temp->next=NULL;

    free(temp);

    return head;

}

**void** display(**struct** node **\*** head)

{

**struct** node **\***temp;

    temp=head;

    while(temp!=NULL)

    {

        cout<<temp->data<<" ";

        temp=temp->next;

    }

    cout<<"\n";

}

**int** main()

{

**int** choice;

**struct** node **\***head;

    head=NULL;

    do

    {

        cout<<"1.Add\n2.Display\n3.Delete\n4.Exit\nEnter Your Choice:";

        cin>>choice;

        switch(choice)

        {

            case 1: head=add(head);

                    break;

            case 2: display(head);

                    break;

            case 3:head=remove(head);

                    break;

            case 4: break;

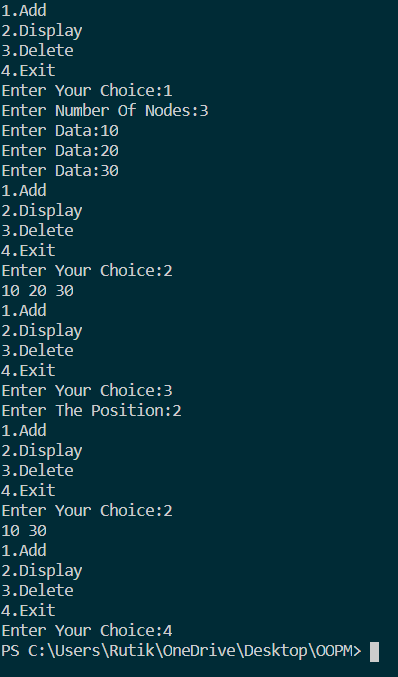
        }

    }while(choice!=4);

    return 0;

}

Output For Q1.:



Source Code For Q2.:

#include <iostream>

using **namespace** std;

**struct** time

{

**int** hours,mins,secs;

};

**void** addsecs(**struct** time **&**t, **struct** time **&**s)

{

    s.secs+= 60 \* 60 \* t.hours + 60 \* t.mins + t.secs;

}

**void** convert(**struct** time **&**s)

{

    s.hours = s.secs / (60 \* 60);

    s.mins = (s.secs - (60 \* 60 \* s.hours)) / 60;

    s.secs = (s.secs - (60 \* 60 \* s.hours) - (60 \* s.mins));

}

**int** main()

{

**struct** time t1,t2,s;

    cout << "TIME 1\nEnter Number of Hours:";

    cin >> t1.hours;

    cout << "Enter Number of Mins:";

    cin >> t1.mins;

    cout << "Enter Number of Seconds:";

    cin >> t1.secs;

    cout << "TIME 2\nEnter Number of Hours:";

    cin >> t2.hours;

    cout << "Enter Number of Mins:";

    cin >> t2.mins;

    cout << "Enter Number of Seconds:";

    cin >> t2.secs;

    addsecs(t1,s);

    addsecs(t2,s);

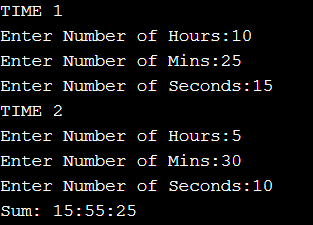
    convert(s);

    cout << "Sum: " << s.hours << ":" << s.mins << ":" << s.secs;

    return 0;

}

Output For Q2.:



**Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On time Completion and Submission (2)** | **Knowledge of the topic (4)** | **Implementation and Output (4)** | **Total (10)** |
|  |  |  |  |

**Date & Signature of teacher:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Rutik Yerunkar | | |
| **Lab Experiment No.** | 10 | **Roll No.** | 8851 |
| **Date Of Perf.:** | 30-09-20 | **Date Of Sub.:** | 29-11-20 |
| **Expt. Title** | Study of polymorphism using operator overloading | | |
| **CO Mapping** | ECL305.3 | | |

**Problem Definition:**

Write a C++ program that uses an overloaded == operator to compare two strings

.**Objective of the Experiment:** Understanding the concept of operator overloading.

|  |
| --- |
| **Attach the Source code and output:** |

Source Code:

#include<iostream>

#include<cstring>

using **namespace** std;

**class** Exp10

{

**private:**

**char** a[100];

**public:**

            Exp10()

            {

                cout<<"Enter A String:";

                cin.getline(a,100);

            }

**void** show()

            {

                cout<<a<<endl;

            }

**bool** operator ==(Exp10 e)

            {

                if(strcmp(a,e.a)==0)

                    return 1;

                else

                    return 0;

            }

};

**int** main()

{

    Exp10 e1,e2;

    if(e1==e2)

        cout<<"String Is Equal\n";

    else

             cout<<"String Are Not Equal\n";

    return 0;

}

Output:





|  |
| --- |
| The program was tested for different sets of inputs.  Program is working is SATISFACTORY NOT SATISFACTORY ( Tick appropriate outcome) |

**Post Lab Questions:**

1. Write a C++ program that uses an overloads unary operator '-' to negate a number.

2. Write a program that uses an overloaded += operator .This operator should allow statements like

s1 += s2;

where s2 is added (concatenated) to s1 and the result is left in s1. The operator should also permit the results of the operation to be used in other calculations, as in s3 = s1 += s2;

.

Source Code For Q1.:

#include<iostream>

using **namespace** std;

**class** Negate

{

**private:**

**int** x, y;

**public:**

**void** get()

            {

                cout<<"Enter numbers:";

                cin>>x>>y;

            }

**void** display()

            {

                cout<<x<<" "<<y;

            }

**void** operator-()

            {

                x=-x;

                y=-y;

            }

};

**int** main()

{

    Negate n;

    n.get();

    cout<<"Numbers are :";

    n.display();

    -n;

    cout<<"\nNegated numbers are :";

    n.display();

    return 0;

}

Output For Q1.:



Source Code For Q2.:

#include <iostream>

#include <cstring>

using **namespace** std;

**class** String

{

**private:**

**enum** { SZ = 80 };*//size of String objects*

**char** str[SZ];

**public:**

    String()

    {

        strcpy(str," ");

    }

    String( **char** s[] )

    {

        strcpy(str, s);

    }

**void** display()

    {

        cout << str;

    }

**void** operator += (String ss)

    {

        if( strlen(str) + strlen(ss.str) >= SZ )

        {

            cout <<"\nString overflow";

        }

        strcat(str, ss.str);

    }

};

**int** main()

{

    String s1 = "Hey ";

    String s2 = "Buddy";

    String s3;

    cout <<"\ns1=";

    s1.display();

    cout <<"\ns2=";

    s2.display();

    s1 += s2;

    s3=s1;

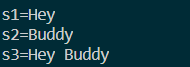
    cout <<"\ns3="; s3.display();

    cout << endl;

    return 0;

}

Output For Q2.:



**Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On time Completion and Submission (2)** | **Knowledge of the topic (4)** | **Implementation and Output (4)** | **Total (10)** |
|  |  |  |  |

**Date & Signature of teacher:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Student** | Rutik Yerunkar | | |
| **Lab Experiment No.** | 11 | **Roll No.** | 8851 |
| **Date Of Perf.:** | 20-10-20 | **Date Of Sub.:** | 29-11-20 |
| **Expt. Title** | Study of inheritance | | |
| **CO Mapping** | ECL305.3 | | |

**Problem Definition:**

Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float). Each of these three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display its data. Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata(), and then displaying the data with putdata().

.**Objective of the Experiment:** Understanding the concept of inheritance

|  |
| --- |
| **Attach the Source code and output:** |

Source Code:

#include<iostream>

#include<string>

using **namespace** std;

**class** publication

{

**private:**

    string title;

**float** price;

**public:**

**void** getdata()

    {

        cout<<"enter title"<<endl;

        cin>>title;

        cout<<"enter price"<<endl;

        cin>>price;

    }

**void** putdata()

    {

        cout<<"Title:"<<title<<endl<<"Price:"<<price<<endl;

    }

};

**class** book:**public** publication

{

**private:**

**int** pagecnt;

**public:**

**void** getdata()

    {

        publication::getdata();

        cout<<"Pagecount:"<<endl;

        cin>>pagecnt;

    }

**void** putdata()

    {

        publication::putdata();

        cout<<"Pagecount:"<<pagecnt<<endl;

    }

};

**class** tape:**public** publication

{

**private:**

**float** ptime;

**public:**

**void** getdata()

    {

        publication::getdata();

        cout<<"Enter playtime:"<<endl;

        cin>>ptime;

    }

**void** putdata()

    {

        publication::putdata();

        cout<<"Play time:"<<ptime<<endl;

    }

};

**int** main()

{

    book b;

    tape t;

    b.getdata();

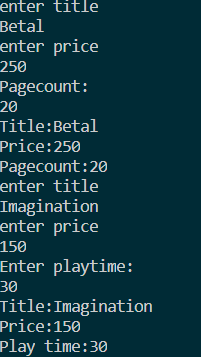
    b.putdata();

    t.getdata();

    t.putdata();

}

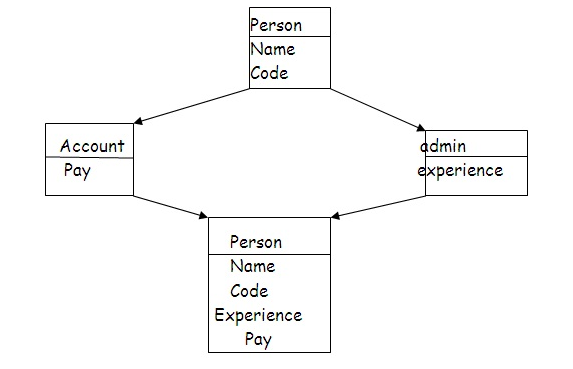
Output:



|  |
| --- |
| The program was tested for different sets of inputs.  Program is working is SATISFACTORY NOT SATISFACTORY ( Tick appropriate outcome) |

**Post Lab Questions:**

1. Consider a class network of the following figure. The class master derives information from both account and admin classes which in turn derives information from the class person. Define all the four classes and write a program to create, update and display the information contained in master objects



.

Source Code:

#include<iostream>

#include<string>

using **namespace** std;

**class** person

{

**private:**

    string name;

**int** code;

**public:**

**void** getdata()

    {

        cout<<"enter name"<<endl;

        cin >>name;

        cout<<"enter code"<<endl;

        cin>>code;

    }

**void** putdata()

    {

        cout<<"Name:"<<name<<endl<<"Code:"<<code<<endl;

    }

};

**class** account:**public** person

{

**private:**

**int** pay;

**public:**

**void** getdata()

    {

        person::getdata();

        cout<<"Payment:"<<endl;

        cin>>pay;

    }

**void** putdata()

    {

        person::putdata();

        cout<<"Payment:"<<pay<<endl;

    }

};

**class** admin:**public** person

{

**private:**

    string experience;

**public:**

**void** getdata()

    {

        cout<<"Experience:"<<endl;

        cin>>experience;

    }

**void** putdata()

    {

        cout<<"Experience:"<<experience<<endl;

    }

};

**class** master : **public** account, **public** admin

{

**public:**

**void** getdata()

        {

            account::getdata();

            admin::getdata();

        }

**void** putdata()

        {

            account::putdata();

            admin::putdata();

        }

};

**int** main()

{

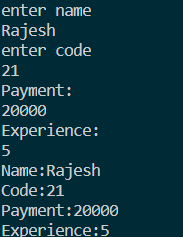
    master m;

    m.getdata();

    m.putdata();

}

Output:



**Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On time Completion and Submission (2)** | **Knowledge of the topic (4)** | **Implementation and Output (4)** | **Total (10)** |
|  |  |  |  |

**Date & Signature of teacher:**