

OBJECT ORIENTED PROGRAMMING LAB

LAB RECORD

Submitted by

TANISHQ AGARWAL

Submitted to: Dr. Kunj Bihari Meena



2021-2022

Department of Computer Science & Engineering
JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY,
AB ROAD, RAGHOGARH, DT. GUNA-473226 MP, INDIA

Table of Contents

Lab Exercise with topic	Page No.
Lab Exercise 1: Revisiting C	3-12
Lab Exercise 2: Revisiting C	13-18
Lab Exercise 3: Function Overloading	19-29
Lab Exercise 4: Classes and Friend Functions	30-35
Lab Exercise 5: Constructors and Destructors	36-40
Lab Exercise 6: Operator Overloading	41-46
Lab Exercise 7: Inheritance	47-54
Lab Exercise 8: Run time Polymorphism	55-57
Lab Exercise 9: File Handling	58-68

Lab Exercise 1: Revisiting C

Q1. WAP to find the number of digits in a given number?

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>

using namespace std;

int digitcount() //function declaration and definition
{
    int n;
    int count=0;
    cout<<"Please enter the number whose digits are to be counted:";
    cin>>n;
    while(n!=0){
        n=n/10;
        count++;
    }
    cout<<"The number of digits in the given number is equal to:"<<count;
    return 0;
}

int main()
{
    digitcount(); //function calling
    return 0;
}

```

Q2. WAP to find the factorial of a given number using recursion?

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

```

```

#include <iostream>

using namespace std;

int factorial(int num)
{
    int fact=1;
    if(num!=0){
        fact=num*factorial(num-1);
    }
    else{
        fact= fact*1;
    }
    return fact;
}

int main()
{
    int n=0;
    int result;
    cout<<"Enter the number whose factorial is to be calculated:";
    cin>>n;
    result=factorial(n);
    cout<<"The factorial of "<n<<" is: "<result;
    return 0;
}

```

Q3. WAP to print “Hello JUET!” without main() function?

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>
# define q3 cout<<"Hello JUET!" // MACRO DEFINITION

using namespace std;

int question3() //function declaration and definition
{
    if(q3){

    }
}

```

```

return 0;
}

int main() //main function declaration
{
    question3(); // function calling

return 0;
}

```

Q4. WAP to print “Hello JUET!” without using any semicolon?

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>

using namespace std;

int main()
{
    if (cout<<"Hello JUET!") {

    }

return 0;
}

```

Q5. WAP to round off an integer “i” to the next largest multiple of another integer “j”. For example, you will get 259 if i=256 is rounded off to the next largest multiple of j=7.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>

using namespace std;

int question5() //function declaration and definition
{
    int a,b,c=0;
    cout<<"Enter the numbers A and B:";

```

```

cin>>a>>b;
for(int i=1;i<=a;i++){
    c=b*i;
    if(c>=a){
        cout<<"The multiple is:"<<c;
        break;
    }
    else{

    }
}
return 0;
}

int main()
{
    question5(); //function calling
    return 0;
}

```

Q6. WAP which finds a four-digit number AABB which is a perfect square. A and B represent different digits. For example: 7744 is a four-digit perfect square number which is also satisfying the condition AABB ie. first two digits (AA=77) are same and last two digits (BB=44) are same.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>
#include <math.h>

using namespace std;

int question6() //function declaration and definition
{
    float numf,sq;
    int num,i=0;
    int t1,t2=0;
    int a[4];
    cout<<"Enter the number u want to check\n";
    cin>>numf;
    sq=sqrt(numf);
    num=sq;
    if(num==sq){

```

```

        cout<<"The number entered is a perfect square number.\n";
        t1=1;
    }
    else{
        cout<<"The number entered is not a perfect square number.\n";
        t1=0;
    }
    for(i=0;i<=3;i++){
        a[i]=num%10;
        num=num/10;
    }
    for(i=3;i>=0;i--){
        a[3-i]=a[i];
    }
    if((a[0]==a[1]) && (a[2]==a[3])){
        cout<<"The number entered is symmetric.\n";
        t2=1;
    }
    else{
        cout<<"The number is asymmetric.\n";
        t2=0;
    }
    if(t1==1 && t2==1){
        cout<<"The number is of required type.\n";
    }
    else{
        cout<<"The number is not of required type..";
    }
    return 0;
}

```

```

int main(){
    question6(); // function calling
    return 0;
}

```

Q7. Write a function which takes a string as input from user and returns the length of that string without using any string library functions. Call this function from main function.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

```

```

#include <iostream>

```

```

using namespace std;

```

```

int question7() //function declaration and definition
{
    string ar;

    int count=0,i=0;

    cout<<"Please enter the array of characters into this array:";

    cin>>ar;

    while(ar[i]!='\0'){

        count++;

        i++;

    }

    cout<<"The length of the string is: "<<count;

    return 0;

}

int main(){ //driver main function

    question7(); //function calling

    return 0;

}

```

Q8. Write a function strcat(s,t) which concatenates the string t to the end of string s. Call this function from main function.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>

using namespace std;

int strcat() //function declaration and definition
{

```



```

string a;

string b;

cout<<"Please enter the value of string A:";

cin>>a;

cout<<"Please enter the value of string B:";

cin>>b;

cout<<"The combined and concatenated final string is: "<<a + b;

return 0;

}

int main(){ //main function

strcat(); //function calling

return 0;

}

```

Advanced Practice Problems:

Q1. Given an array A of size N-1 and given that there are numbers from 1 to N with one element missing; Write program to find the missing number.

Test case 1: Given array: 1 2 3 5; missing element is 4.

Test case 2: Given array: 1 2 3 4 5 6 7 8 10; missing element is 9.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>

#include <stdlib.h>

using namespace std;

int app1() //function declaration and definition

{

```

```

int n=0;
int orig=0;
cout<<"Please enter the no of characters in the array with the missing element:";
cin>>n;
int *a=(int *)malloc(n*sizeof(int));
for(int i=0;i<=n;i++){
    cout<<"Enter the value for position "<<i<<"in the array:\n";
    cin>>a[i];
}
for(int i=0;i<=n;i++){
    if(a[i+1]-a[i]==1){

    }
    else{
        cout<<"The missing element is: "<<a[i]+1;
        break;
    }
}
return 0;
}
int main(){
    app1(); //function calling
    return 0;
}

```

Q2. Write the function strend(s,t), which returns 1 if the string t occurs at the end of the string s, and zero otherwise.

Sample Test case1:

Input:

s="Object Oriented Programming using C++"

t="Using C++"

Output: 1

Sample Test case2:

Input:

s="Object Oriented Programming using C++"

t="Programming"

Output: 0

```
/**
 *
 */
```

```
//This program is developed by Tanishq Agarwal (Er. No:211B326)
```

```
/**
 *
 */
```

```
#include <iostream>
```

```
#include <string.h>
```

```
using namespace std;
```

```
int strend(string s,string t) //function declaration and definition
```

```
{
```

```
int tr=0;
```

```
int index=s.find(t);
```

```
int ls=s.length();
```

```
if(s.find(t)!= string::npos){
```

```
    if(index>(ls/2)){
```

```
        cout<<"1";
```

```
    }
```

```
else{
```

```
    cout<<"0";
```

```
}
```

```
}  
return 0;  
  
}  
  
int main() //driver main function  
{ string s1,s2;  
cout<<"Please enter the string s:";  
getline(cin,s1);  
cout<<"Please enter the string t:";  
getline(cin,s2);  
strend(s1,s2); //function calling  
return 0;  
}
```

Lab Exercise 2: Revisiting C

Q1. Write a function that finds the minimum and the maximum value in an array of N integers. Inputs to the function are the array of integers, an integer variable containing the length of the array and pointers to integer variables that will contain the minimum and the maximum values. The function prototype is:

void minmax(int array[], int length, int * min, int * max);

Write a main function that uses this function to find and display the minimum and the maximum values of an array of integers.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>

using namespace std;

void minmax(int array[],int length,int *minq,int *maxq)    //function declaration and definition
{
    *maxq=array[0];
    for(int i=0;i<=length;i++){
        if(array[i]>(*maxq)){
            *maxq=array[i];
        }
    }
    *minq=array[0];
    for(int i=0;i<=length;i++){
        if(array[i]<(*minq)){
            *minq=array[i];
        }
    }
    return ;
}

int main()    //Driver Main function
```

```

{
int len,minq,maxq=0;
cout<<"Enter an array of integers..\n";
cout<<"Enter the array size first:";
cin>>len;
int *n= new int[len];
for(int i=0;i<=len;i++){
    cout<<"Enter element at position"<<i<<":\n";
    cin>>n[i];
}
minmax(n,len,&minq,&maxq);
cout<<"The maximum value in the array is:"<<maxq;
cout<<"The minimum values is the array is:"<<minq;
} return 0;
}

```

Q2. Write a program to generate random numbers in given range [m, n].

Test case :

Input: m=10, n=50

Output: 34

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>

using namespace std;

int question2(int l,int u){
srand((unsigned)time(NULL));
for(int i=0;i<10;i++){
    int random = 1 + (rand()%((u-l)+1));
    cout<<random<<endl;
}
return 0;
}

int main()    //main function declaration
{
    cout<<"Set lower and upper limits of the random number generator profile:";
    cin>>a>>b;
    question2(a,b); // function calling
}

```

```
return 0;
}
```

Q3. Write a function to reverse an array of long double types. Call this function from main function.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>

using namespace std;

int question3(long double ar[],int sz)          // function declaration and definition
{
    long double ar2[sz];
    int j=0;
    for(j=0;j<sz;j++){
        ar2[j]=ar[sz-j];
    }
    for(j=0;j<sz;j++){
        cout<<"The element at the position "<<j<<" in the new array is:"<<ar2[j]<<"\n";
    }
    return 0;
}

int main()          // Driver main() Function
{
    cout<<"Enter size of the long double array:";
    int sz=0;
    cin>>sz;
    long double *test=new long double[sz];
    for(int x=0;x<sz;x++){
        cout<<"Enter the value of original array for position "<<x<<"\n";
        cin>>test[x];
    }
    question3(test,sz);
    return 0;
}

```

Q4. Write a program to perform the addition of two matrices.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)

```

```

//*****

#include <iostream>

using namespace std;
int question4()          // function declaration and definition
{
    int len=0;
    int i,j=0;
    cout<<"Enter the order of A and B matrix to be added:";
    cin>>len;
    int a[len][len];
    int b[len][len];
    int sum[len][len];
    for(i=0;i<len;i++){
        for(j=0;j<len;j++){
            cout<<"Enter the value for A matrix at position A "<<"["<<i<<"]"<<" "<<"["<<j<<"]";
            cin>>a[i][j];
        }
    }

    for(i=0;i<len;i++){
        for(j=0;j<len;j++){
            cout<<"Enter the value for B matrix at position B "<<"["<<i<<"]"<<" "<<"["<<j<<"]";
            cin>>b[i][j];
        }
    }

    for(i=0;i<len;i++){
        for(j=0;j<len;j++){
            sum[i][j]=a[i][j]+b[i][j];
        }
    }

    for(i=0;i<len;i++){
        for(j=0;j<len;j++){
            cout<<"The value for SUM matrix at position SUM "<<"["<<i<<"]"<<"
"<<"["<<j<<"]"<<"is:"<<sum[i][j]<<"\n";
        }
    }

    return 0;
}

int main()      // Driver main() Function
{

```



```

question4();
return 0;
}

```

Advanced Practice Problems:

Q1. Write a C++ program to find the highest occurring digit in prime numbers in a given range.

Given a range L to R, the task is to find the highest occurring digit in prime numbers lie between L and R (both inclusive). If multiple digits have same highest frequency print the largest of them. If no prime number occurs between L and R, output -1.

Examples:

Input : L = 1 and R = 20.

Output : 1

Prime number between 1 and 20 are 2, 3, 5, 7, 11, 13, 17, 19.

1 occur maximum i.e 5 times among 0 to 9.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>

using namespace std;

int app1() // function declaration and definition
{
    int i,a,b,arcnt=0,ctr=0,maxo;
    int count[10]={0,0,0,0,0,0,0,0,0,0};
    int pn[10];
    cout<<"Enter the values for upper limit and lower limit for check:";
    cin>>a>>b;
    for(i=a;i<=b;i++){
        for(int j=1;j<=b;j++){
            if(i%j==0){
                ctr++;
            }
        }
    }
    if(ctr<=2){
        pn[arcnt]=i;
    }
}

```

```

        arcnt++;
    }
    ctr=0;
}
for(i=0;i<arcnt;i++) {
    while(pn[i]>0) {
        count[pn[i]%10]++;
        pn[i]=pn[i]/10;
    }
}
maxo=count[0];
for(i=0;i<10;i++) {
    if(count[i]>maxo)
        maxo=count[i];
}
for(i=9;i>=0;i--) {
    if(count[i]==maxo) {
        cout << "Digit: " << i << endl;
        break;
    }
}
return 0;
}

int main()    // Driver main() Function
{
    app1(); //function calling
    return 0;
}

```

Lab Exercise 3: Function Overloading

Q1. Write C++ Program to swap two variable using reference variables.

```
/**
//This program is developed by Tanishq Agarwal (Er. No:211B326)
**/

#include <iostream>

using namespace std;

int question1(int &x,int &y){
int temp=x;
x=y;
y=temp;
return 0;
}

int main()
{
int x,y;
cout<<"ENTER X AND Y:";
cin>>x>>y;
question1(x,y);
cout<<"SWAPPED X:"<<x<<endl<<"SWAPPED Y:"<<y<<endl;
return 0;
}
```

Q2 Write a function that finds the minimum and the maximum value in an array of N integers. Inputs to the function are the array of integers, an integer variable containing the length of the array and references to integer variables that will contain the minimum and the maximum values. The function prototype is:

void minmax(int array[], int length, int& min, int & max);

```
/**
//This program is developed by Tanishq Agarwal (Er. No:211B326)
**/

#include <iostream>
using namespace std;
void minmax(int array[],int length,int &minq,int &maxq)
{
```

```

maxq=array[0];
for(int i=0;i<=length;i++){
    if(array[i]>(maxq)){
        maxq=array[i];
    }
}
minq=array[0];
for(int i=0;i<=length;i++){
    if(array[i]<(minq)){
        minq=array[i];
    }
}
}
}
Int main(){
int len,minq,maxq=0;
    cout<<"Enter an array of integers..\n";
    cout<<"Enter the array size first:";
    cin>>len;
    int *n= new int[len];
    for(int i=0;i<=len;i++){
        cout<<"Enter element at position"<<i<<":\n";
        cin>>n[i];
    }
    minmax(n,len,minq,maxq);
    cout<<"The maximum value in the array is:"<<maxq;
    cout<<"The minimum values is the array is:"<<minq;
return 0;
}

```

Q.3 Create a four-function calculator for fractions. Here are the formulas for the four arithmetic operations applied to fractions:

Addition: $a/b + c/d = (a*d + b*c) / (b*d)$

Subtraction: $a/b - c/d = (a*d - b*c) / (b*d)$

Multiplication: $a/b * c/d = (a*c) / (b*d)$

Division: $a/b / c/d = (a*d) / (b*c)$

The user should type the first fraction (two values a and b), an operator, and a second fraction (two values c and d). The program should then display the results in fraction ie. (Numerator/ denominator).

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>

```

```

using namespace std;
int add(int &a,int &b,int &c,int &d)
{
    int r_nm,r_dnm=0;
    r_nm=(a*d)+(b*c);
    r_dnm=(b*d);
    cout<<"RESULT:"<<r_nm<<"/"<<r_dnm<<endl;
    return 0;
}
int sub(int &a,int &b,int &c,int &d)
{
    int r_nm,r_dnm=0;
    r_nm=(a*d)-(b*c);
    r_dnm=(b*d);
    cout<<"RESULT:"<<r_nm<<"/"<<r_dnm<<endl;
    return 0;
}
int mult(int &a,int &b,int &c,int &d)
{
    int r_nm,r_dnm=0;
    r_nm=(a*c);
    r_dnm=(b*d);
    cout<<"RESULT:"<<r_nm<<"/"<<r_dnm<<endl;
    return 0;
}
int div(int &a,int &b,int &c,int &d)
{
    int r_nm,r_dnm=0;
    r_nm=(a*d);
    r_dnm=(b*c);
    cout<<"RESULT:"<<r_nm<<"/"<<r_dnm<<endl;
    return 0;
}
int main()
{
    int a,b,c,d,ops=0;
    cout<<"Enter the first fraction numerator(A):";
    cin>>a;
    cout<<"Enter the first fraction denominator(B):";
    cin>>b;
    cout<<"Enter the second fraction numerator(C):";
    cin>>c;
    cout<<"Enter the second fraction denominator(D):";
    cin>>d;
    cout<<"1. ADDITION"<<endl<<"2. SUBTRACTION"<<endl<<"3.
MULTIPLICATION"<<endl<<"4. DIVISION"<<endl;

```

```

cin>>ops;
switch(ops){
case 1: add(a,b,c,d);
        break;
case 2: sub(a,b,c,d);
        break;
case 3: mult(a,b,c,d);
        break;
case 4: div(a,b,c,d);
        break;
default: cout<<"ERROR!! Enter a correct menu item.."<<endl;
        break;
}
return 0;
}

```

Q4. Create a class rectangle with attributes length and width. Provide member functions that calculate the perimeter and area of the rectangle. Provide member functions to get the values from users and display the values of member variables. Write a program to test the class.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;
class rectangle{
    float length,width=0;
public:
    void insertdataset(float l,float w){
        length=l;
        width=w;
    }
    void peri(){
        float p=(2*length)+(2*width);
        cout<<"The value of perimeter of the given rectangular dimensions will be:"<<p<<endl;
    }

    void area(){
        float ar=(length*width);
        cout<<"The value of area of the given rectangular dimensions will be:"<<ar<<endl;
    }
};
int main()
{
    rectangle rec;
    float length,width;
    cout<<"Enter the value of length of the rectangle:"<<endl;

```

```

    cin>>length;
    cout<<"Enter the value of width of the rectangle:"<<endl;
    cin>>width;
    rec.insertdataset(length,width);
    rec.peri();
    rec.area();
return 0;
}

```

Q5. Write a function that accepts two arguments: a string name of a movie and an integer running time in minutes. Provide a default value for the minutes so that if you call the function without an integer argument, the minutes default to 90. Write a main() function that proves that you can call the function with a string argument alone as well as with a string and an integer.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;
void question5(string mv_name, int runtime=90){
    cout<<"The movie "<<mv_name<<"is of "<<runtime<<"minutes"<<endl;
}
int main()
{
    string mv;
    int rntm=0;
    cout<<"Enter the movie name:"<<endl;
    cin>>mv;
    cout<<"Enter the running time in minutes:"<<endl;
    cin>>rntm;
    question5(mv,rntm);
    question5(mv);
return 0;
}

```

Q6. Create a class named Shirt that has the public data members collarsize and sleeveLength. Create a class named Pants that has the public data members waistSize and inSeam. Write a program that declares one object of each type Shirt and Pants and assigns values to the objects' data fields. Write two overloaded functions named displayClothingFacts(). One version of the function takes a Shirt object as an argument; the other version takes a Pants object. Each version displays the facts about the piece of clothing. Your main() function should demonstrate that you can call displayClothingFacts() with either type of clothing.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)

```

```

//*****
#include <iostream>
using namespace std;
class shirt{
    public:
    float collarSize=0;
    float sleeveLength=0;
    void sh_ins();
};
void shirt::sh_ins(){
cout<<"ENTER THE SHIRT COLLAR SIZE:"<<endl;
cin>>collarSize;
cout<<"ENTER THE SHIRT SLEEVE LENGTH:"<<endl;
cin>>sleeveLength;
}

class pants{
    public:
    float waistSize=0;
    float inSeam=0;
    void pt_ins();
};
void pants::pt_ins(){
cout<<"ENTER THE PANT WAIST SIZE:"<<endl;
cin>>waistSize;
cout<<"ENTER THE PANT INSEAM LENGTH:"<<endl;
cin>>inSeam;
}

void displayClothingfacts(shirt obj_s){
    cout<<"The collar size of the shirt is:"<<obj_s.collarSize<<endl;
    cout<<"The sleeve length of the shirt is:"<<obj_s.sleeveLength<<endl;
}

void displayClothingfacts(pants obj_p){
    cout<<"The waist size of the pant is:"<<obj_p.waistSize<<endl;
    cout<<"The inseam length of the pant is:"<<obj_p.inSeam<<endl;
}
int main()
{
    shirt s;
    s.sh_ins();
    pants p;
    p.pt_ins();
    displayClothingfacts(s);
    displayClothingfacts(p);
}

```



```
return 0;
}
```

Q7. Define a class named Movie. Include private fields for the title, year, and name of the director. Include three public functions with the prototypes void Movie::setTitle(string); , void Movie::setYear(int); and void Movie::setDirector(string);. Include another function that displays all the information about a Movie. Write a main() function that declares a movie object named myFavoriteMovie. Set and display the object's fields.

```
//*****
```

```
//This program is developed by Tanishq Agarwal (Er. No:211B326)
```

```
//*****
```

```
#include <iostream>
```

```
using namespace std;
```

```
class movie{
```

```
    string title;
```

```
    string nameDirector;
```

```
    int year;
```

```
    public:
```

```
    void setTitle(string);
```

```
    void setYear(int);
```

```
    void setDirector(string);
```

```
    void movieDisplay();
```

```
}typedef myFavoriteMovie,mov;
```

```
void movie::setTitle(string t){
```

```
    cin>>title;
```

```
}
```

```
void movie::setYear(int y){
```

```
    cin>>year;
```

```
}
```

```
void movie::setDirector(string d){
```

```
    cin>>nameDirector;
```

```
}
```

```
void movie::movieDisplay(){
```

```
    cout<<"The Title of the movie is:"<<title<<endl;
```

```
    cout<<"The Director of the movie is:"<<nameDirector<<endl;
```

```
    cout<<"The Release Year of the movie is:"<<year<<endl;
```

```
}
```

```
int main(){
```

```
    movie myFavoriteMovie;
```

```
    string temp;
```

```
    int y_temp;
```

```
    cout<<"Enter the movie title:"<<endl;
```

```

cin>>temp;
myFavoriteMovie.setTitle(temp);
cout<<"Enter the movie release year:"<<endl;
cin>>y_temp;
myFavoriteMovie.setYear(y_temp);
cout<<"Enter the Director of the movie:"<<endl;
cin>>temp;
myFavoriteMovie.setDirector(temp);
myFavoriteMovie.movieDisplay();
return 0;
}

```

Q8. Write a class definition for an order class for a nightclub that contains a table number, a server's name, and the number of patrons at the table. Include a private static data member for the table minimum charge, which is \$4.75. Write a main() function that declares no object of order class type, but uses a static member function to display the table minimum charge.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;
class order{
public:
int tabno=0;
string svname;
int pat=0;
static float minchg;
public:
static float print(){
return minchg;
}
};
float order::minchg=4.75;
int main(){
float res=order::print();
cout<<"The minimum charges for a table is:"<<res<<endl;
return 0;
}

```

ADVANCED PRACTISE PROBLEM:

Q1. An automobile company has serial number for engine parts starting from AA0 to FF9. The other characteristics of parts to be specified in a structure are: Year of manufacturing, material (Steel, Aluminum, Iron etc.) and quantity manufactured.

Write a C++ program to include following tasks:

- (a) Specify a C structure that includes four members viz. serial numbers, year of manufacturing, material type, and quantity. Declare an array of structure of size 60.
- (b) Automatically initialize the values of structure members as follows :
- i. Generate serial numbers such that first part has serial number AA0 and 60th part has serial number FF9.
 - ii. Randomly generate year of manufacturing in the range [1990-2018]
 - iii. Randomly initialize material type from three choices ie. Steel, Aluminum, Iron
 - iv. Randomly initialize quantity in the range [1-1000]
- (c) Display the information of the parts with serial numbers between any given range such as [BB1,CC6].

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
#include<time.h>
using namespace std;
struct vehicle{
    int year=0;
    int quantity=0;
    string mat[3]={ "STEEL","IRON","ALUMINIUM"};
    char serial[3]={'A','A','0'};
};
int main(){
    struct vehicle vhec[61];
    srand((unsigned)time(NULL));
    for(int i=1;i<=60;i++){
        vhec[i].year= 1990+(rand()%28);
        vhec[i].quantity= 1+(rand()%1000);
        static int j=0;
        switch(i){
            case 1 ... 10:vhec[i].serial[0]=vhec[i].serial[1]='A';
                if(j<=9){
                    vhec[i].serial[2]=(vhec[i].serial[2]) + j;
                    j++;
                }
                break;

```

```

case 11 ... 20:vhec[i].serial[0]=vhec[i].serial[1]='B';
    static int j=0;
    if(j<=9){
        vhec[i].serial[2]=(vhec[i].serial[2]) + j;
        j++;
    }
    break;
case 21 ... 30:vhec[i].serial[0]=vhec[i].serial[1]='C';
    static int k=0;
    if(k<=9){
        vhec[i].serial[2]=(vhec[i].serial[2]) + k;
        k++;
    }
    break;
case 31 ... 40:vhec[i].serial[0]=vhec[i].serial[1]='D';
    static int l=0;
    if(l<=9){
        vhec[i].serial[2]=(vhec[i].serial[2]) + l;
        l++;
    }
    break;
case 41 ... 50:vhec[i].serial[0]=vhec[i].serial[1]='E';
    static int m=0;
    if(m<=9){
        vhec[i].serial[2]=(vhec[i].serial[2]) + m;
        m++;
    }
    break;
case 51 ... 60:vhec[i].serial[0]=vhec[i].serial[1]='F';
    static int n=0;
    if(n<=9){
        vhec[i].serial[2]=(vhec[i].serial[2]) + n;
        n++;
    }
    break;
default: cout<<"Hello world!!";
    break;
}
}
for(int x=1;x<61;x++){
    cout<<"The vehicle no is:"<<x<<endl;
    cout<<"The vehicle's serial no is:"<<vhec[x].serial<<endl;
    cout<<"The vehicle's year of manufacture is: "<<vhec[x].year<<endl;
    cout<<"The vehicle's quantity is : "<<vhec[x].quantity<<endl;
    cout<<"The vehicle's material is : "<<vhec[x].mat[0+(rand()%2)]<<endl;

```

```

        cout<<endl;
    }

    //REVERSE SEARCH

    string range_l;
    string range_u;
    cout<<"ENTER A RANGE TO REVERSE SEARCH FOR:"<<endl;
    cout<<"ENTER LOWER RANGE LIMIT:"<<endl;
    cin>>range_l;
    cout<<"ENTER UPPER RANGE LIMIT:"<<endl;
    cin>>range_u;
    int l_f=range_l[0];
    int l_l=range_l[2];
    l_l=l_l-'0';
    int u_f=range_u[0];
    int u_l=range_u[2];
    u_l=u_l-'0';
    for(char z=l_f;z<=u_f;z++){
        cout<<"The vehicle no is:"<<l_l+1<<endl;
        cout<<"The vehicle's serial no is:"<<vhc[l_l+1].serial<<endl;
        cout<<"The vehicle's year of manufacture is: "<<vhc[l_l+1].year<<endl;
        cout<<"The vehicle's quantity is : "<<vhc[l_l+1].quantity<<endl;
        cout<<"The vehicle's material is : "<<vhc[l_l+1].mat[0+(rand()%2)]<<endl;
        cout<<endl;
        if(l_l+1<=9)
        {l_l++;
        }
        else if(l_l>9){
            l_l=0;
        }
        else{
        }
    }
    return 0;
}

```

Lab Exercise 4: Classes and Friend Functions

Q1. The annual examination results of 10 students are tabulated as follows:

Roll No.	Subject 1	Subject 2	Subject 3
.	.	.	.
.	.	.	.
.	.	.	.

Write a program to read the data and determine the following:

- Total marks obtained by each student.**
- The highest marks in each subject and the Roll No. of the students who secured it.**
- The student who obtained the highest total marks**

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;

class student{
    int sub1=0,sub2=0,sub3=0;
    int rn=0;
    int total=0;
public:
    void setdata(int roll,int sb1,int sb2,int sb3){
        rn=roll;
        sub1=sb1;
        sub2=sb2;
        sub3=sb3;
    }
    void totalmarks(student obj[],int sz){
        for(int i=0;i<sz;i++){
            obj[i].total=obj[i].sub1 + obj[i].sub2 + obj[i].sub3;
            cout<<"The total marks of student having roll no:"<<obj[i].rn<<"is:"<<total<<endl;
        }
    }

    void highestmarks(student obj[],int sz){
        int high1=obj[0].sub1;
        int high2=obj[0].sub2;
        int high3=obj[0].sub3;
    }
}
```

```

for(int i=0;i<sz;i++){
    if(obj[i].sub1 < obj[i+1].sub1){
        high1=obj[i+1].sub1;
        cout<<"The highest scoring student in subject 1 is roll no:"<<obj[i+1].rn<<endl;
    }
}

for(int i=0;i<sz;i++){
    if(obj[i].sub2 < obj[i+1].sub2){
        high2=obj[i+1].sub2;
        cout<<"The highest scoring student in subject 2 is roll no:"<<obj[i+1].rn<<endl;
    }
}

for(int i=0;i<sz;i++){
    if(obj[i].sub3 < obj[i+1].sub3){
        high3=obj[i+1].sub3;
        cout<<"The highest scoring student in subject 3 is roll no:"<<obj[i+1].rn<<endl;
    }
}

}

void topper(student obj[],int sz){
    int top=obj[0].total;
    for(int i=0;i<sz;i++){
        if(obj[i].total < obj[i+1].total){
            top=obj[i+1].total;
        }
    }
}

};

int main(){
    student s[10],s1;
    int subject1=0,subject2=0,subject3=0;
    for(int j=0;j<10;j++){
        cout<<"Enter marks of subject 1 for student"<<j+1<<endl;
        cin>>subject1;
        cout<<"Enter marks of subject 2 for student"<<j+1<<endl;
        cin>>subject2;
        cout<<"Enter marks of subject 3 for student"<<j+1<<endl;
        cin>>subject3;
        s[j].setdata(j+1,subject1,subject2,subject3);
    }
}

```

```

    }
    cout<<"ALL DATA IS SET SUCCESSFULLY!!"<<endl;

    s1.totalmarks(s,10);
    s1.highestmarks(s,10);
    s1.topper(s,10);
return 0;
}

```

Q2. Define a class to represent a bank account, including the following data members:

- Name of the depositor
- Account number
- Type of account
- Balance amount in the account

And member functions:

- To assign initial values
- To deposit an amount
- To withdraw an amount after checking the balance
- To display the name and balance

Write a main program to test the program.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;

class bank{
    string name;
    int acno;
    string type;
    long int acbal;
public:
    void setdata(){
        acno=1;
        type="SAVINGS";
        acbal=0;
    }
    void deposit(bank obj[],int sz){
        int dep=0;
        for(int i=0;i<sz;i++){
            cout<<"Enter the name of bank account holder:"<<endl;
            cin>>obj[i].name;
            cout<<"Enter the account number of bank account holder:"<<endl;

```



```

cin>>obj[i].acno;
cout<<"Enter the type of bank account:"<<endl;
cin>>obj[i].type;
cout<<"Enter the amount to be deposited:"<<endl;
cin>>dep;
obj[i].acbal=obj[i].acbal+dep;
cout<<"AMOUNT DEPOSITED SUCCESSFULLY AND BALANCE UPDATED!!"<<endl;
}
}

void checkandwithdraw(bank obj[]){
int tempacno=0,withdraw=0;
char ch;
cout<<"Enter the account number of which balance is to be checked:"<<endl;
cin>>tempacno;
cout<<"The account balance for given account number is:"<<obj[tempacno-1].acbal<<endl;
A:
cout<<"DO YOU WANT TO WITHDDRAW AMOUNT??"<<endl;
cin>>ch;
if(ch=='Y'|| ch=='y'){
    cout<<"Enter amount to withdraw:"<<endl;
    cin>>withdraw;
    cout<<"PROCESSING..."<<endl;
    if(ac)
        obj[tempacno-1].acbal=obj[tempacno-1].acbal-withdraw;
    cout<<"WITHDRAWAL SUCCESSFUL!!"<<endl<<"NEW BALANCE
IS:"<<obj[tempacno].acbal<<endl;
}
else if(ch=='N'||ch=='n'){
    cout<<"SURE!!";
    cout<<endl;
}
else{
    cout<<"Enter a correct character:"<<endl;
    goto A;
}
}

void showdetails(bank obj[]){
int tempacno;
cout<<"Enter the account no for which details are required:"<<endl;
cin>>tempacno;
cout<<"The name of the requested user is:"<<obj[tempacno-1].name<<" "<<"and his/her
account balance is:"<<obj[tempacno-1].acbal<<endl;
}
};

```

```

int main(){
    int usersize;
    cout<<"Enter number of users"<<endl;
    cin>>usersize;
    bank user[usersize],user1;
    user1.setdata();
    user1.deposit(user,usersize);
    user1.checkandwithdraw(user);
    user1.showdetails(user);
return 0;
}

```

Q3. Create two classes DM and DB which store the value of the distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimeters depending on the object on display.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;

class DB;
class DM{
float met=0;
float cent=0;
friend float adddmdb(DM,DB);
public:
    void setdatadm(){
        cout<<"ENTER VALUE FOR METRES:"<<endl;
        cin>>met;
        cout<<"ENTER VALUE FOR CENTIMETRES:"<<endl;
        cin>>cent;
    }
};

class DB{
float feet=0;

```

```

float inches=0;
friend float adddmdb(DM,DB);
public:
    void setdatadb(){
        cout<<"ENTER VALUE FOR FEET:"<<endl;
        cin>>feet;
        cout<<"ENTER VALUE FOR INCHES:"<<endl;
        cin>>inches;
    }
};

float adddmdb(DM obj1,DB obj2){
    int inp=0;
    float res=0;
    cout<<"Input result format:"<<"1. METRES"<<endl<<"2. FOOT"<<endl<<"3.
INCHES"<<endl<<"4.CENTIMETRES"<<endl;
    cin>>inp;
    switch(inp){
        case 1: res=(obj1.cent/100)+obj1.met+(obj2.feet*0.304)+(obj2.inches*0.0254);
            break;
        case 2: res=(obj1.cent*0.032)+(obj1.met*100*0.032)+obj2.feet+(obj2.inches*0.083);
            break;
        case 3: res=(obj1.met*39.36)+((obj1.cent/100)*39.36)+obj2.inches+(obj2.feet*12);
            break;
        case 4: res=obj1.cent+(obj1.met*100)+(obj2.feet*30.48)+(obj2.inches*2.54);
            break;
        default: cout<<"Input format correctly!"<<endl;
            break;
    }
    return res;
}

int main(){
    DM d1;
    DB d2;
    d1.setdatadm();
    d2.setdatadb();
    float converted=adddmdb(d1,d2);
    cout<<"The required conversion is: "<<converted<<endl;
    return 0;
}

```

Lab Exercise 5: Constructors and Destructors

Q1. Write a program to perform addition of two complex numbers using constructor overloading. The first constructor which takes no argument is used to create objects which are not initialized, second which takes one argument is used to initialize real and imaginary parts to equal values and third which takes two argument is used to initialize real and imaginary to two different values. Include sum member function that takes two objects and performs the addition of these two objects. Write display function to display the object in complex form i.e. $2+i6$. Write main function to test your program.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;

class complex{
    int real;
    int img;

public:

    void sum(complex,complex);
    void display();
    complex(){
        complex c1,c2,c3;
    }
    complex(int init){
        real=init;
        img=init;
    }
    complex(int RL, int IMG){
        real=RL;
        img=IMG;
    }
};

void complex::sum(complex a,complex b){
    real=a.real + b.real;
    img=a.img + b.img;
    cout<<"The sum of these complex numbers is:"<<real<<"+"<<img<<endl;
}

void complex::display(){
    cout<<"The complex number for selected object entered is:"<<real<<"+"<<img<<endl;
}

```

```

int main(){
    complex c1(3);
        complex c2(2,4);
        c1.display();
        c2.display();
        complex c3(0);
        c3.sum(c1,c2);
return 0;
}

```

Q2. A point in a two-dimensional plane having coordinate as (x,y), can be represented by a class whose private data members are x and y. Write the constructor and destructor functions of the class. The constructor should initialise (x,y) by passing parameters values. Now, a rectangle can be represented by the top-left and bottom-right vertices. Define a class say 'Rectangle' whose private data members are top-left and bottom-right vertices. Write the parameterised constructor function of the class 'Rectangle'. Also, write the destructor function. Finally, write a program to show the order in which different constructors and destructors are called.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;

class cartesian{
    int x,y;
public:
    cartesian(){
        cout<<"Cartesian Constructor 1 called"<<endl;
        x=0;
        y=0;
    }
    cartesian(int a,int b){
        cout<<"Cartesian Constructor 2 called"<<endl;
        x=a;
        y=b;
    }
    ~cartesian(){
        cout<<"Cartesian class Destructor Called!!"<<endl;
    }
};

class rectangle{
    int tp1,tp2,bt1,bt2;

```

```

public:
    rectangle(int tp1,int tp2,int bt1,int bt2){
        cout<<"Rectangle Constructor 1 called"<<endl;
        cartesian p1(tp1,tp2);
        cartesian p2(bt1,bt2);
        cout<<"TOP coordinates:"<<tp1<<tp2<<endl;
        cout<<"BOTTOM coordinates:"<<bt1<<bt2<<endl;
    }
    ~rectangle(){
        cout<<"Rectangle class Destructor Called!!"<<endl;
    }
};

int main(){
    rectangle r1(2,3,4,1);
    return 0;
}

```

Q3. Write a program to show that, the constructor and destructor functions of a globally declared object are the first and last functions, respectively to be called in a program.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;

class question3{
    int i,j;
public:
    void setdata();
    void displaydata();
    question3(){
        cout<<"Constructor for global obj called!"<<endl;
        i=1;j=1;
        cout<<"i="<<i<<"j="<<j<<endl;
    }
    ~question3(){
        cout<<"Destructor for global obj called !!"<<endl;
        i=0;j=0;
        cout<<"i="<<i<<"j="<<j<<endl;
    }
};

void question3::setdata(){
    cout<<"Enter value for i and j:"<<endl;
}

```

```

        cin>>i>>j;
    }
    void question3::displaydata(){
        cout<<"i and j : "<<i<<" "<<j<<endl;
    }
    question3 q1;

    int main(){
        q1.displaydata();
        q1.setdata();
    return 0;
    }

```

Q4. Write a program to show that constructors follow the property of function overloading as well as default parameter. Also, show that in case of constructor also, default parameter may create problem in implementing function overloading.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;

class question4{
    float l;
public:
    question4(){
        cout<<"First Constructor called!!"<<endl;
        l=1;
    }
    question4(int t){
        cout<<"Second Constructor called!!"<<endl;
        l=t;
    }
    question4(float x){
        cout<<"Third Constructor called!!"<<endl;
        l=x;
    }
    question4(int x,int y=2){
        cout<<"Fourth Constructor called!!"<<endl;
        l=x+y;
    }
    void displaydata();
};

void question4::displaydata(){
    cout<<"Value of l after constructor call is: "<<l<<endl;
}

```

```
}  
  
int main(){  
    question4 q;  
    q.displaydata();  
    //question4 w(5); //Error causing even when default parameter exists!!  
    //w.displaydata();  
    question4 e((float)4.9);  
    e.displaydata();  
    question4 f(3,8);  
    e.displaydata();  
  
    return 0;  
}
```


Lab Exercise 6: Operator Overloading

Q1. Write Design a class Distance that includes following data members: feet, inches. It has the following member function:-

- **Constructor**, that initializes the distance to 0,0 by default.
- **Parameterized Constructor** that initializes the distance with some values; give a check so that the inches part is always less than 12.0.
- **Display function**
- **Overloaded – operator** to subtract 2 distances
- **Overloaded + operator** to add 2 distances
- **Overload += and -= operator**
- **Overload > and < operators** to compare two distance

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;
class distance1{

    int feet,inch;

public:
    distance1(){
        feet=0;
        inch=0;
    }

    distance1(int f,int i){
        feet=f;
        inch=i;
        if(inch>=12){
            f++;
            inch=0;
        }
        else{
        }
    }

    void display(){
        cout<<"The distance is:"<<feet<<"feet and "<<inch<<"inches."<<endl;
    }
};
```

```

    }

    distance1 operator + (distance1 &obj){
        distance1 oba;
        oba.feet= feet+obj.feet;
        oba.inch= inch+obj.inch;
        return oba;
    }

    distance1 operator - (distance1 &obj){
        distance1 obs;
        obs.feet= feet+obj.feet;
        obs.inch= inch+obj.inch;
        return obs;
    }

    bool operator > (distance1 &obj){
        if(feet>obj.feet){
            cout<<"Object 1 has greater distance in feet!"<<endl;
            return 1;
        }
        else{
            cout<<"Object 2 has greater distance in feet!"<<endl;
            return 0;
        }
    }

    bool operator < (distance1 &obj){
        if(feet<obj.feet){
            cout<<"Object 1 has lesser distance in feet!"<<endl;
            return 1;
        }
        else{
            cout<<"Object 2 has lesser distance in feet!"<<endl;
            return 0;
        }
    }
};

int main(){
    distance1 d1(5,5);
    distance1 d2(5,3);
    distance1 da = d1 + d2;
    distance1 ds = d1 - d2;
    da.display();
}

```

```

        ds.display();

        if(d1 > d2){
            cout<<"The distance d1 is greater than d2!"<<endl;
        }
        else{
            cout<<"The distance d2 is greater than d1!"<<endl;
        }

        if(d1 < d2){
            cout<<"The distance d1 is lesser than d2!"<<endl;
        }
        else{
            cout<<"The distance d2 is lesser than d1!"<<endl;
        }

    return 0;
}

```

Q2. Create a class rational for performing arithmetic with fractions. Use an integer variable to represent the private data of the class-the numerator and denominator. Provide a member function to get input from the user. This function should also check that denominator entered is not 0, if it is zero print invalid input. Provide a function to display the values. Overload +, -, *, / operators to add, subtract, multiply and divide the objects of this class.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;

class rational{
    int num;
    int den;
public:
    rational(){
        num=1;
        den=1;
    }

    rational(int n, int d){
        num=n;
        den=d;
    }

    void setdata(){
        cout<<"Enter a value for fractions numerator:"<<endl;
        cin>>num;
    }
}

```

```

cout<<"Enter a value for fractions denominator:"<<endl;
cin>>den;
if(den==0){
    cout<<"FRACTION INVALID!!\n Enter Denominator again:"<<endl;
    cin>>den;
}
else{
    cout<<"Fraction entered successfully!"<<endl;
}
}

void display()const{
cout<<"Entered fraction is:"<<num<<"|"<<den<<endl;
}

rational operator +(rational obj){
    int sum_num= (this->num * obj.den)+(obj.num * this->den);
    int sum_den= (this->den * obj.den);
    cout<<"The sum of these fractions is:"<<sum_num<<"|"<<sum_den<<endl;
    return rational(sum_num,sum_den);
}

rational operator -(rational obj){
    int dif_num= (this->num * obj.den)- (obj.num * this->den);
    int dif_den= (this->den * obj.den);
    cout<<"The difference of these fractions is:"<<dif_num<<"|"<<dif_den<<endl;
    return rational(dif_num,dif_den);
}

rational operator *(rational obj){
    int pro_num= (this->num * obj.num);
    int pro_den= (this->den * obj.den);
    cout<<"The product of these fractions is:"<<pro_num<<"|"<<pro_den<<endl;
    return rational(pro_num,pro_den);
}

rational operator /(rational obj){
    int div_num= (this->num * obj.den);
    int div_den= (obj.num * this->den);
    cout<<"The division result of these fractions is:"<<div_num<<"|"<<div_den<<endl;
    return rational(div_num,div_den);
}

};

int main(){

```

```

rational r1;
r1.setdata();
r1.display();
rational r2;
r2.setdata();
r2.display();
rational r3;
r3=r1+r2;
r3=r1-r2;
r3=r1*r2;
r3=r1/r2;
return 0;
}

```

Q3. Overload '+' operator that adds two strings to make a third string. Write a program to do the following tasks:

- i. Create uninitialized string objects**
- ii. Creates the objects with string constants.**
- iii. Concatenates two strings properly using operator overloading.**
- iv. Displays a desired string object**

```

#include <iostream>
using namespace std;
class add
{
string a;
public:
void setData()
{
getline(cin,a);
}
add operator +(add& y)
{ add c;
c.a = a+ y.a;
return c;
}
void display()
{ cout<<a;
}
};
int main()
{
add x,y,z;

```

```
cout<<"enter --> ";  
x.setData();  
cout<<"enter --> ";  
y.setData();  
z = x+y;  
z.display();  
return 0;  
}
```

TANISHQ AGARWAL(211B326)

Lab Exercise 7: Inheritance

Q1. Consider an example of declaring the examination result. Design three classes: Student, Exam, and Result. The Student class has data members representing roll number, name. Create the class Exam by inheriting Student class. The Exam class adds fields (data members) representing the marks scored in six subjects. Derive the Result from the Exam class, and it has its own fields such as total_marks. Write an interactive program to model this relationship.

```
//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;

class student{
    int roll=0;
    string name;
public:
    student(){
        name="";
        roll++;
    }

    void datainsert(){
        cout<<"Enter the roll number:"<<endl;
        cin>>roll;
        cout<<"Enter name of student : "<<endl;
        cin>>name;
    }

    void studisplay() const{
        cout<<"Student's Name:"<<name<<endl<<"Student's roll number:"<<roll<<endl;
    }

};

class exam: public student{
public:
    int marks[6];
    void examinsert(){
        for(int i=0;i<6;i++){
            cout<<"Enter marks for subject "<<i<<endl;
            cin>>marks[i];
        }
    }
}
```

```

void examdisplay()const{
    cout<<"Student's marks:"<<endl;
    for(int i=0;i<6;i++){
        cout<<"Marks for subject "<<i<<"are:"<<marks[i];
    }
}
};

class result: public exam{
int total_marks;
public:
    void resultdisplay(){
        studisplay();
        examdisplay();
        for(int i=0;i<6;i++){
            total_marks+=marks[i];
        }
        cout<<"The total marks of the student is:"<<total_marks;
    }
};

int main(){
    student stuobj;
    stuobj.datainsert();
    exam exobj;
    exobj.examinert();
    result resobj;
    resobj.resultdisplay();
    return 0;
}

```

Q2. There is a class student, that stores name of school or university from which he is enrolled and name of highest degree he has obtained so far. It has the function to get and display the members. Design a class Employee with name and employee number. Derive Manager, Scientist and Laborer classes from Employee class. The manager class has extra attribute title (string type) and dues (float type). The scientist class has extra attributes number of publications. The Laborer class has nothing extra. The classes have necessary functions for set and display the information. The manager and scientist are students of a university also. Use inheritance. Test your program by creating objects of type manager, scientists and laborer.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>

```



```

using namespace std;

class Student{
char school[100];
char degree[100];
public:
    void getdata(){
        cout<<"Enter the name of the School/University:"<<endl;
        cin>>school;
        cout<<"Enter the name of the highest degree obtained until now:"<<endl;
        cin>>degree;
    }

    void display()const{
        cout<<"The name of the School/University:"<<school<<endl;
        cout<<"The name of the highest degree obtained until now:"<<degree<<endl;
    }
};

class Employee{
char empname[50];
int empno;
public:
    void getemp(){
        cout<<"Enter Employee's name:"<<endl;
        cin>>empname;
        cout<<"Enter employee number:"<<endl;
        cin>>empno;
    }

    void dispemp() const{
        cout<<"The name of the employee is: "<<empname<<"and the employee number
is:"<<empno<<endl;
    }
};

class Manager: public Employee,public Student{
    string title;
    float dues;
public:
    void getman(){
        cout<<"Enter the title of the Manager:"<<endl;
        cin>>title;
        cout<<"Enter the amount of dues if any or enter zero:"<<endl;
        cin>>dues;
    }
};

```

```

    }

    void mandisplay()const{
        display();
        dispemp();
        cout<<"The title of the Manager:"<<title<<endl;
        cout<<"The amount of dues is:"<<dues<<endl;
    }

};

class Scientist: public Employee,public Student{
    int no_of_pub;
public:
    void getsci(){
        cout<<"Enter the no of publications:"<<endl;
        cin>>no_of_pub;
    }

    void scidisplay()const{
        display();
        dispemp();
        cout<<"The no of publications by the scientist is:"<<no_of_pub<<endl;
    }
};

class Laborer: public Employee{
    char labname[100];
public:
    void getlab(){
        cout<<"Enter the name of the laborer:"<<endl;
        cin>>labname;
    }

    void labdisplay()const{
        dispemp();
        cout<<"The name of teh laborer is:"<<labname<<endl;
    }
};

int main(){
    Student Stuobj;
        Stuobj.getdata();
        Employee empobj;
        empobj.getemp();
        Manager manobj;

```

```

        manobj.getman();
        Scientist sciobj;
        sciobj.getsci();
        Laborer labobj;
        labobj.getlab();
        manobj.mandisplay();
        sciobj.scidisplay();
        labobj.labdisplay();
return 0;
}

```

Q3. Write a program with a mother class and a derived daughter class. Both of them should have a method void display () that prints a message (different for mother and daughter).In the main function declare an object of class daughter and call the display() method on it. Also suitably invoke the display() function of mother class using this object of class daughter.

```

/*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
*****/
#include <iostream>
using namespace std;

class mother{
public:
    void display(){
        cout<<"Mother class invoked!!"<<endl;
    }
};

class daughter: public mother{
public:
    void display(){
        cout<<"Daughter class invoked!!"<<endl;
    }
};

int main(){
    daughter dobj;
    dobj.display();
    dobj.mother::display();
return 0;
}

```

Q4. An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in Fig.1. The figure also shows the minimum information required for each class. Specify all

the classes and define methods to create the database and retrieve individual information as and when required.

```
/**
//This program is developed by Tanishq Agarwal (Er. No:211B326)
**/

#include <iostream>
using namespace std;
class staff{
char code;
char name[50];
public:
    void getstaff(){
        cout<<"Enter the staff name:"<<endl;
        cin>>name;
        cout<<"Enter staff code:"<<endl;
        cin>>code;
    }
    void dispstaff(){
        cout<<"The staff name is:"<<name<<endl;
        cout<<"The staff code is : "<<code<<endl;
    }
};

class teacher:public staff{
char subject[20];
int publication;
public:
    void getteacher(){
        cout<<"Enter the teacher subject:"<<endl;
        cin>>subject;
        cout<<"Enter no of publications by the teacher:"<<endl;
        cin>>publication;
    }
    void showteacher(){
        cout<<"The teacher's subject is:"<<subject<<endl;
        cout<<"The no of publication by the teacher is : "<<publication<<endl;
    }
};

class typist:public staff{
float speed;
public:
    void gettypist(){
        cout<<"Enter the speed of the typist"<<endl;
        cin>>speed;
    }
};
```

```

    }
    void showtypist(){
        cout<<"The speed of the typist is: "<<speed<<endl;
    }
};

class officer:public staff{
    char grade[10];
public:
    void getofficer(){
        cout<<"Enter the grade of the ranking officer:"<<endl;
        cin>>grade;
    }
    void showofficer(){
        cout<<"The grade of the raking officer is: "<<grade<<endl;
    }
};

class regular:public typist{
    float wage;
public:
    void getregular(){
        cout<<"Enter the wage of the regular typist:"<<endl;
        cin>>wage;
    }
    void showregular(){
        cout<<"The daily wage of the regualr category typist is:"<<wage<<endl;
    }
};

class casual:public typist{
    float daily_wages;
public:
    void getcasual(){
        cout<<"Enter daily wages of the casual typist :"<<endl;
        cin>>daily_wages;
    }
    void showcasual(){
        cout<<"The daily wages of the casual typist is:"<<daily_wages<<endl;
    }
};

int main(){
    staff staffobj;
    staffobj.getstaff();
    teacher teachobj;

```

```
    teachobj.getteacher();  
    typist typeobj;  
    typeobj.gettypist();  
    officer offobj;  
    offobj.getofficer();  
    regular regobj;  
    regobj.getregular();  
    casual casobj;  
    casobj.getcasual();  
return 0;  
}
```

Lab Exercise 8: Inheritance

Q1. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initialize baseclass data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements.

(A). Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area using the concept of dynamic binding. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles and as base and height in the case of triangles and used as follows:

Area of rectangle = $x * y$

Area of triangle = $\frac{1}{2} * x * y$

(B). Extend the Program-1 to display the area of circle. This requires addition of a new derived class 'circle' that computes the area of a circle. Remember, for a circle we need only one value, its radius, but the get_data() function in the base class requires two values to be passed. (Hint: Make the second argument of get_data() function as a default one with zero value.)

(C). Run the above program with the following modifications:

- Remove the definition of display_area() from one of the derived classes.
- In addition to the above change, declare the display_area() as pure virtual in the base class shape.

Comment on the output in each case.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
using namespace std;

class shape{
public:
    double x;
    double y;

    void get_data(){
        cout<<"Enter first dimension of figure accessed:"<<endl;
        cin>>x;
        cout<<"Enter second dimension of figure accessed:"<<endl;
        cin>>y;
    }
};
```

```

    }

    virtual void display_area(){
        cout<<"Base class display_area() accessed!!"<<endl<<"No area can be presented for the
base class!!"<<endl;
    }
    //virtual void display_area()= 0;

};

class triangle:public shape{
public:
    void display_area(){
        double areatr=0.5*x*y;
        cout<<"The area of triangle is:"<<areatr<<endl;
    }
};

class rectangle:public shape{
public:
    void display_area(){
        double arearec=x*y;
        cout<<"The area of rectangle is:"<<arearec<<endl;
    }
};

class circle:public shape{
public:
    void display_area(){
        if(y==x){
            double areacir=3.14*x*y;
            cout<<"The area of rectangle is:"<<areacir<<endl;
        }
        else{
            cout<<"Different values of radius detected!!"<<endl;
        }
    }
};

int main()
{
    shape *shptr;
    triangle tr;
    rectangle rec;
    circle cr;
    shptr = &tr;

```



```
shptr->get_data();
shptr->display_area();
shptr=&rec;
shptr->get_data();
shptr->display_area();
shptr= &cr;
shptr->get_data();
shptr->display_area();
return 0;
}
```

TANISHQ AGARWAL(211B326)

Lab Exercise 9: File Handling

Q1. Write a program that creates a text file “TEXT.txt” on the disk. Write text on this file. Read this file and display the following information on the screen in two columns:

- 1. Number of lines**
- 2. Number of words**
- 3. Number of characters**

Strings should be left-justified and numbers should be right-justified in a suitable field width. Also handle the error by displaying suitable error message.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
#include <fstream>
#include <bits/stdc++.h>
#include <iomanip>
using namespace std;

int main(){
    fstream fout;
    fout.open("TEXT.txt");
    if(fout.is_open()){
        fout<<"Hello"<<" "<<"1234"<<" "<<"!@#%^&*()"<<endl;
        fout<<"My name is Tanishq Agarwal aka 047_Pegasus!!"<<endl;
        cout<<"Data Written successfully to the file!!"<<endl;
    }
    else{
        cout<<"ERROR OPENING FILE TO WRITE DATA!!";
        exit(0);
    }
    fout.close();
    fstream fin;
    fin.open("TEXT.txt");
    int nline=0;
    int nwords=0;
    int nchar=0;
    if(fin.is_open()){
        char line[100];
        if(fin.eof()!=1){
            fin.getline(line,100);
            for(int i=0;fin.eof()!=1;i++){
                if(line[i]!=' ' && line[i]!='\n'){
                    nchar++;
                }
            }
        }
    }
}
```

```

        else if(line[i]==' '){
            nwords++;
        }
        else if(line[i]=='\n'){
            nline++;
            nwords++;
        }
        else{

        }
    }
}
cout<<"No of lines:"<<nline<<endl;
cout<<"No of words:"<<nwords<<endl;
cout<<"No of characters:"<<nchar<<endl;
}

else{
    cout<<"ERROR OPENING FILE TO READ DATA!!";
    exit(0);
}
fin.close();
return 0;
}

```

Q2. Write a program to read the file and store the lines into an array. Also handle the error by displaying suitable error message.

```

//*****
//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
#include <fstream>
#include <bits/stdc++.h>
#include <iomanip>
using namespace std;

int main(){
    . char temp[500];
    int i=0;
    char ch;
    fstream fin;
    fin.open("MYFILE.txt");
    if(fin.good()){
        cout<<"The file is opened successfully!!"<<endl;
    }
}

```

```

else{
    cout<<"FILE OPENING ERROR!!"<<endl;
    exit(0);
}
fin.seekg(0);
while(fin.eof()!=1){
    fin.get(ch);
    temp[i]=ch;
    i++;
}
i=0;
cout<<"The file content read is:"<<endl;
while(temp[i]!='\0'){
    cout<<temp[i];
    i++;
}
return 0;
}

```

Q3. Write a program to copy a file in another name. Also handle the error by displaying suitable error message.

```

//*****

//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>
#include <fstream>
#include <bits/stdc++.h>
#include <iomanip>
using namespace std;

int main(){
    char c;

    ifstream fin1;
    fin1.open("FIRST.txt",ios::in);
    fin1.seekg(0);
    if(fin1.good()){
        cout<<"FILE 1 OPENED SUCCESSFULLY FOR COPYING!!"<<endl;
    }
    else{
        cout<<"ERROR OPENING COPYING FILE 1 !!"<<endl;
        exit(0);
    }

    ofstream fin2;
    fin2.open("SECOND.txt",ios::out);

```

```

    fin2.seekp(0);
    if(fin2.good()){
        cout<<"FILE 2 OPENED SUCCESSFULLY FOR COPYING!!"<<endl;
    }
    else{
        cout<<"ERROR OPENING COPYING FILE 2 !!"<<endl;
        exit(0);
    }

    cout<<"Both files opened successfully!!"<<endl<<"Beginning Copying!!"<<endl;

    while(fin1.eof()!=1){
        fin1.get(c);
        fin2<<c;
    }
    fin1.close();
    fin2.close();
    cout<<"Data copied Successfully!!"<<endl;

return 0;
}

```

Q4. Write a program to merge two files and write it in a new file. Also handle the error by displaying suitable error message.

```

//*****

//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>
#include <fstream>
#include <bits/stdc++.h>
#include <iomanip>
using namespace std;

int main(){
    char chr;

    ifstream f1;
    f1.open("Q4FIRST.txt");
    if(f1.good()){
        cout<<"File 1 OPENED SUCCESSFULLY IN READ MODE!!"<<endl;
    }
    else{
        cout<<"ERROR OPENING FILE 1!!"<<endl;
        exit(0);
    }

    ifstream f2;

```

```

f2.open("Q4SECOND.txt");
if(f2.good()){
    cout<<"File 2 OPENED SUCCESSFULLY IN READ MODE!!"<<endl;
}
else{
    cout<<"ERROR OPENING FILE 2!!"<<endl;
    exit(0);
}

ofstream f3;
f3.open("Q4THIRD.txt",ios::ate);
if(f3.good()){
    cout<<"File 3 OPENED SUCCESSFULLY in WRITE MODE!!"<<endl;
}
else{
    cout<<"ERROR OPENING FILE 3!!"<<endl;
    exit(0);
}

while(f1.eof()!=1){
    f1.get(chr);
    f3<<chr;
}
f3.seekp(0,ios::end);
f3<<endl;
while (f2.eof()!=1){
    f2.get(chr);
    f3<<chr;
}

cout<<"Data merged and Copied to third file successfully!!"<<endl;

f1.close();
f2.close();
f3.close();

return 0;
}

```

Q5. Write a program to encrypt and decrypt a text file. Also handle the error by displaying suitable error message.

```

//*****

//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>
#include <fstream>

```

```

#include <bits/stdc++.h>
#include <iomanip>
using namespace std;

int main(){
char echar;
    char dchar;
    char opt;
    int r;
    cout<<"Enter program in which mode:"<<"1:Encryption
mode"<<endl<<"2.Decryption Mode"<<endl;
    cin>>r;
    if(r==1){
        goto E;
    }
    else if(r==2){
        goto D;
    }
    else{
        cout<<"Enter a proper mode!";
        exit(0);
    }
    E:{
        fstream encrypt_fin;
        encrypt_fin.open("SAMPLEQ5.txt");
        if(encrypt_fin.good()){
            cout<<"File opened successfully!!"<<endl<<"Encryption Operation
started!!"<<endl;
        }
        else{
            cout<<"File not found so not opened!! EXITING PROGRAM"<<endl;
            exit(0);
        }
        while(encrypt_fin.eof()!=1){
            encrypt_fin.get(echar);
            if(echar==' '){
                encrypt_fin.seekp(0,ios::cur);
                encrypt_fin.put('$');
            }
            else if(echar>'b'&& echar<'x'){
                encrypt_fin.seekp(0,ios::cur);
                encrypt_fin.put((char)((int)echar+1));
            }
            else if(echar>'B'&& echar<'X'){
                encrypt_fin.seekp(0,ios::cur);
                encrypt_fin.put((char)((int)echar-1));
            }
        }
    }
}

```

```

    }
    else if(echar>='2' && echar<='7'){
        encrypt_fin.seekp(0,ios::cur);
        encrypt_fin.put(echar+1);
    }
    else{
    }
}

encrypt_fin.close();
}
cout<<"File encrypted successfully!"<<endl<<"If u wish to decrypt it enter D or
d:"<<endl;
cin>>opt;
if(opt=='d' || opt=='D'){
    D:{
        fstream decrypt_fin;
        decrypt_fin.open("SAMPLEQ5.txt");
        if(decrypt_fin.good()){
            cout<<"File opened successfully!!"<<endl<<"Decryption Operation
started!!"<<endl;
        }
        else{
            cout<<"Decryption failed as file linking failed!! EXITING PROGRAM"<<endl;
            exit(0);
        }
        while(decrypt_fin.eof()!=1){
            decrypt_fin.get(dchar);
            if(dchar=='$'){
                decrypt_fin.seekp(0,ios::cur);
                decrypt_fin.put(' ');
            }
            else if(dchar>'b' && dchar<'x'){
                decrypt_fin.seekp(0,ios::cur);
                decrypt_fin.put((char)((int)echar-1));
            }
            else if(dchar>'B' && dchar<'X'){
                decrypt_fin.seekp(0,ios::cur);
                decrypt_fin.put((char)((int)echar+1));
            }
            else if(dchar>='2' && dchar<='7'){
                decrypt_fin.seekp(0,ios::cur);
                decrypt_fin.put(echar-1);
            }
            else{
            }
        }
    }
}

```



```

    }
    cout<<"File Decrypted Successfully!!"<<endl<<"Now file can be read
normally!!"<<endl;
    decrypt_fin.close();
}

else{
    cout<<"File remains encrypted forever now!!"<<endl;
    exit(0);
}

return 0;
}

```

Q6. Write the sorted list of integers in two files named ‘Source1’ and ‘Source2’. Write a program that reads the contents of both the files and stores the merged list in sorted form in a new file named ‘Target’. Also handle the error by displaying suitable error message.

```

//*****

//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****

#include <iostream>
#include <fstream>
#include <bits/stdc++.h>
#include <iomanip>
using namespace std;

int main(){
    int num=0;
    int pos=0;
    int sortnum[10];

    ifstream source1;
    source1.open("Source1.txt");
    if(source1.good()){
        cout<<"Source1 file is opened successfully!!"<<endl;
    }
    else{
        cout<<"File 1 not available!!"<<endl<<"Program terminating!!"<<endl;
        exit(0);
    }

    ifstream source2;
    source2.open("Source2.txt");
    if(source1.good()){
        cout<<"Source2 file is opened successfully!!"<<endl;
    }
}

```

```

    }
    else{
        cout<<"File 2 not available!!"<<endl<<"Program terminating!!"<<endl;
        exit(0);
    }

    cout<<"Both files found!"<<endl<<"Proceeding operations.."<<endl;

    while(source1.eof()!=1){
        source1>>num;
        sortnum[pos]=num;
        pos++;
    }
    while(source2.eof()!=1){
        source2>>num;
        sortnum[pos]=num;
        pos++;
    }

    cout<<"Beginning Sorting on numbers obtained!"<<endl;
    int sizesort=sizeof(sortnum) / sizeof(sortnum[0]);
    sort(sortnum,(sortnum+sizesort));
    cout<<"Sorting completed successfully!!"<<endl;

    pos=0;

    fstream t;
    t.open("Target.txt");
    if(t.good()){
        cout<<"Target file found."<<endl;
    }
    else{
        cout<<"Target file not found!"<<endl<<"Program Terminating!!"<<endl;
        exit(0);
    }
    while(sortnum[pos]!='\0'){
        t<<sortnum[pos];
        pos++;
    }
    cout<<"Target file written successfully!!"<<endl;
    source1.close();
    source2.close();
    t.close();

return 0;
}

```

Q7. In a loop, prompt the user to enter name data consisting of a first name, middle initial, last name, and employee number (type unsigned long). Then, using formatted I/O with the insertion (<<) operator, write these four data items to a file using an ofstream object. Don't forget that strings must be terminated with a space or other whitespace character. When the user indicates that no more name data will be entered, close the ofstream object, open an ifstream object, read and display all the data in the file, and terminate the program

```
//*****

//This program is developed by Tanishq Agarwal (Er. No:211B326)
//*****
#include <iostream>
#include <fstream>
#include <bits/stdc++.h>
#include <iomanip>
using namespace std;

int main(){
char a;
    char fname[100];
    char mname[100];
    char lname[100];
    unsigned long empno=0;
    ofstream outques7;
    outques7.open("QUEST7.txt");
    if(outques7.good()){
        cout<<"File found. Program continuing!!"<<endl;
    }
    else{
        cout<<"File not found so program terminating!!"<<endl;
        exit(0);
    }
    do{ cout<<"Enter first name:"<<endl;
        cin.getline(fname,100);
        cout<<"Enter middle name:"<<endl;
        cin.getline(mname,100);
        cout<<"Enter last name:"<<endl;
        cin.getline(lname,100);
        cout<<"Enter employee number:"<<endl;
        cin>>empno;
        outques7<<fname<<" "<<setw(5)<<mname<<" "<<setw(5)<<lname<<"
"<<setw(5)<<empno<<endl;
        cout<<"Enter Q to terminate the loop of data entry!!"<<endl;
        cin>>a;
    }while(a!='Q');
    outques7.close();
```

```

cout<<"File closed!!"<<endl;

char read;
ifstream inques7;
inques7.open("QUEST7.txt");
if(inques7.good()){
    cout<<"File found and opened again after writing!! Program continuing!"<<endl;
}
else{
    cout<<"File not found after writing data to it!! Program Terminating!!"<<endl;
    exit(0);
}
while(inques7.eof()!=1){
    inques7.get(read);
    if(read!='\n'){
        cout<<read;
    }
    else{
        cout<<"\n";
    }
}
return 0;
}

```