Q1.

#include<conio.h>

#include<iostream>

using namespace std;

int main(){

int a,b;

cout<<"Enter any two integers\n";

cin>>a>>b;

cout<<"Sum is = "<<a+b<<"\n";

cout<<"Product is = "<<a\*b;

return 0;

}

Q2.

#include<conio.h>

#include<iostream>

using namespace std;

int main(){

int a;

cout<<"Enter a number:\n";

cin>>a;

cout<<"Reversed Number is:\n"<<((a%10)\*10)+(a/10);

return 0;

}

Q3.

#include<iostream>

#include<conio.h>

using namespace std;

int main(){

int n;

float s;

cout<<"Enter the times upto which you want the pattern sum\n";

cin>>n;

for (float i=1;i<=n;i++){

s+=1/i;

}

cout<<"Sum is = "<<s;

return 0;

}

Q4.

#include<iostream>

#include<conio.h>

using namespace std;

int main(){

int n;

float s;

cout<<"Enter the times upto which you want the pattern to be solved\n";

cin>>n;

for (float i=1;i<=n;i++){

if(int(i)%2==0){

s-=i;

cout<<"1\n"<<s;

}

else{

s+=i;

cout<<"2\n"<<s;

}

}

cout<<"Solution is = "<<s;

return 0;

}

Q5.

#include<conio.h>

#include<iostream>

using namespace std;

string palin(string s){

string rs;

for(int i=0;i<s.size();i++){

rs=s[i]+rs;

}

if(rs==s){

return "Palindrome";

}

else{

return "Not Palindrome";

}

}

int main(){

string s;

cout<<"Enter a word:\n";

cin>>s;

cout<<palin(s);

}

Q6.

#include<iostream>

#include<conio.h>

using namespace std;

string primeCheck(int n){

int a;

a=n-1;

while(a>1){

if(n%a==0){

return "Not a prime\n";

}

a--;

}

return "Prime\n";

}

int primeGenerator(){

for(int i=1;i<=100;i++){

int c=0;

for(int j=2;j<i;j++){

if(i%j==0){

c++;

}

}

if(!c){

cout<<i<<" ";

}

}

return 0;

}

int main(){

int n;

cout<<"Enter a number:\n";

cin>>n;

cout<<primeCheck(n);

primeGenerator();

}

Q7.

#include<conio.h>

#include<iostream>

using namespace std;

int main(){

int n;

cout<<"Enter a number:\n";

cin>>n;

cout<<"\n";

for(int i=1;i<=n;i++){

if(n%i==0){

cout<<i<<" ";

}

}

}

Q8.

#include<conio.h>

#include<iostream>

using namespace std;

int main(){

int n;

cout<<"Enter the number of rows you want\n";

cin>>n;

for(int i=0;i<n;i++){

for(int j=0;j<=i;j++){

cout<<"\*";

}

cout<<"\n";

}

}

Q9.

#include<conio.h>

#include<iostream>

using namespace std;

int disp(int &l,int \*arr){

cout<<"Entered array is:\n";

cout<<"[ ";

for(int i=0;i<l;i++){

cout<<arr[i]<<" ";

}

cout<<"]\n\n";

cout<<"Press any key to continue...\n\n";

getch();

return 0;

}

int dispMaxMin(int &l, int \*arr){

int Max=0;

int Min=999999;

for(int i=0;i<l;i++){

if(arr[i]<=Min){

Min=arr[i];

}

else if(arr[i]>=Max){

Max=arr[i];

}

}

cout<<"Minimum number is:"<<Min<<endl;

cout<<"Maximum number is:"<<Max<<endl;

cout<<"Press any key to continue...\n\n";

getch();

return 0;

}

int dispSumAvg(int &l,int \*arr){

int sum=0;

int avg=0;

for(int i=0;i<l;i++){

sum+=arr[i];

}

avg=sum/l;

cout<<"Sum of all elements is:"<<sum<<endl;

cout<<"Average of all elements is:"<<avg<<endl;

cout<<"Press any key to continue...\n\n";

getch();

return 0;

}

int dispRev(int &l,int \*arr){

cout<<"Reversed array is:";

cout<<"[ ";

for(int i=l-1;i>=0;i--){

cout<<arr[i]<<" ";

}

cout<<"]\n\n";

cout<<"Press any key to continue...\n\n";

getch();

return 0;

}

int dispEvenOdd(int &l,int \*arr){

cout<<"Even Numbers are:"<<endl;

for(int i=0;i<l;i++){

if(arr[i]%2==0){

cout<<arr[i]<<" ";

}

}

cout<<endl;

cout<<"Odd Numbers are:"<<endl;

for(int i=0;i<l;i++){

if(!(arr[i]%2==0)){

cout<<arr[i]<<" ";

}

}

cout<<endl;

cout<<"Press any key to continue...\n\n";

getch();

return 0;

}

int main(){

int l;

int choice;

cout<<"Enter the size of array:\n";

cin>>l;

int arr[l];

cout<<"Enter the elements:\n";

for(int i=0;i<l;i++){

cin>>arr[i];

}

do{

cout<<"What do you want to do with this array?\n\n";

cout<<"1. Display maximum and minimum\n";

cout<<"2. Display the sum and average of all elements\n";

cout<<"3. Print Array in reverse\n";

cout<<"4. Print even numbers and odd numbers separately\n";

cout<<"5. Display the array\n";

cout<<"6. Exit\n";

cout<<endl;

cin>>choice;

switch(choice){

case 1:

dispMaxMin(l,arr);

break;

case 2:

dispSumAvg(l, arr);

break;

case 3:

dispRev(l,arr);

break;

case 4:

dispEvenOdd(l,arr);

break;

case 5:

disp(l,arr);

default:

break;

}

}while(choice!=6);

return 0;

}

Q10.

#include<conio.h>

#include<iostream>

#include<cctype>

#include<iomanip>

using namespace std;

int main(){

string s;

string a;

int f;

cout<<"Enter a string:\n";

getline(cin,s);

int p=0;

while(p<s.size()){

s[p]=tolower(s[p]);

for(int i=0;i<a.size();i++){

f=0;

if(s[p]==a[i]){

f=1;

break;

}

}

if(f!=1 && isalpha(s[p])){

a+=s[p];

}

p++;

}

cout << "\tLetter\t\tFrequency\n";

cout << setw(40) << setfill('-') << "" << endl;

for(char i:a){

f=0;

for(char j:s){

if(i==j){

f++;

}

}

cout << "\t "; cout << setw(7) << setfill(' ') << left << i;

cout << " | "; cout << setw(7) << setfill(' ') << right << f << endl;

}

cout<<a;

return 0;

}

Q11.

#include<iostream>

using namespace std;

void Swap(int \*a,int \*b){

int t=\*b;

\*b=\*a;

\*a=t;

return;

}

int main(void){

int a,b;

cout<<"Enter two numbers\n";

cin>>a>>b;

cout<<endl;

cout<<"Numbers before swapping\n";

cout<<"x = "<<a<<endl;

cout<<"y = "<<b<<endl;

Swap(&a,&b);

cout<<"Numbers after swapping\n";

cout<<"x = "<<a<<endl;

cout<<"y = "<<b<<endl;

}

Q12.

#include<conio.h>

#include<iostream>

using namespace std;

int main(){

float sal,da,hra,tax,ta;

cout<<"Enter the salary of the employee\n";

cin>>sal;

cout<<"Enter the Dearness Allowance percentage\n";

cin>>da;

cout<<"Enter the Travelling Allowance\n";

cin>>ta;

cout<<"Enter the House Rent Allowance\n";

cin>>hra;

cout<<"Enter the Income Tax\n";

cin>>tax;

cout<<endl;

da=sal\*(da/100);

ta=sal\*(ta/100);

tax=sal\*(tax/100);

hra=sal\*(hra/100);

cout<<"\tBasic Salary = "<<sal<<endl;

cout<<"\tDA = "<<da<<endl;

cout<<"\tTA = "<<ta<<endl;

cout<<"\tHRA = "<<hra<<endl;

cout<<"---------------------------------------"<<endl;

cout<<"\tAdding all.."<<endl;

cout<<"\tGross Salary = "<<sal+da+ta+hra<<endl;

cout<<"\tDeducting Income Tax.."<<endl;

cout<<"\tNet Salary = "<<(sal+da+ta+hra)-tax<<endl;

return 0;

}

Q13.

#include<iostream>

#include<conio.h>

using namespace std;

void dispAddr(string s1,string s2){

cout<<"Address of characters of string 1"<<endl;

for(int i=0;i<s1.size();i++){

cout<<"\t"<<s1[i]<<" --> "<<(void \*)&s1[i]<<endl;

}

cout<<"\nAddress of characters of string 2"<<endl;

for(int i=0;i<s2.size();i++){

cout<<"\t"<<s2[i]<<" --> "<<(void \*)&s2[i]<<endl;

}

cout<<"Press any key to continue...\n";

getch();

return;

}

void concatWithoutStrcat(string s1,string s2){

cout<<"String 1 --> "<<s1<<endl;

cout<<"String 2 --> "<<s2<<endl;

cout<<"Concatenated string --> "<<s1+s2;

cout<<"Press any key to continue...\n";

getch();

return;

}

void comString(string s1,string s2){

if(s1==s2){

cout<<"Both strings are equal"<<endl;

}

else if(s1>s2){

cout<<"String 1 is greater than string 2"<<endl;

}

else if(s2>s1){

cout<<"String 2 is greater than string 1"<<endl;

}

cout<<"Press any key to continue...\n";

getch();

return;

}

void dispVowels(string s1,string s2){

cout<<"Vowels in string 1 --> "<<endl;

for(int i=0;i<s1.size();i++){

if(s1[i]=='a' || s1[i]=='e' || s1[i]=='i' || s1[i]=='o' || s1[i]=='u'){

cout<<s1[i]<<" ";

}

}

cout<<endl;

cout<<"Vowels in string 2 --> "<<endl;

for(int i=0;i<s2.size();i++){

if(s2[i]=='a' || s2[i]=='e' || s2[i]=='i' || s2[i]=='o' || s2[i]=='u'){

cout<<s2[i]<<" ";

}

}

cout<<endl;

cout<<"Press any key to continue...\n";

getch();

return;

}

void revString(string s1,string s2){

cout<<"Reversed String 1 --> "<<endl;

for(int i=s1.size()-1;i>=0;i--){

cout<<s1[i];

}

cout<<endl;

cout<<"Reversed string 2 --> "<<endl;

for(int i=s2.size()-1;i>=0;i--){

cout<<s2[i];

}

cout<<endl;

cout<<"Press any key to continue...\n";

getch();

return;

}

int main(){

string s1,s2;

int choice;

cout<<"Enter string 1"<<endl;

cin>>s1;

cout<<"Enter string 2"<<endl;

cin>>s2;

do{

cout<<endl;

cout<<"What do you want to do with these strings\n";

cout<<"1. Show Address of all the alphabets in strings\n";

cout<<"2. Concatenate both strings without using strcat()\n";

cout<<"3. Display all vowels in both strings\n";

cout<<"4. Compare both strings\n";

cout<<"5. Reverse the strings\n";

cout<<"6. Exit\n";

cin>>choice;

switch(choice){

case 1:

dispAddr(s1,s2);

break;

case 2:

concatWithoutStrcat(s1,s2);

break;

case 3:

dispVowels(s1,s2);

break;

case 4:

comString(s1,s2);

break;

case 5:

revString(s1,s2);

break;

default:

break;

}

}while(choice!=6);

}

Q14.

#include<conio.h>

#include<iostream>

using namespace std;

void fibboSeries(){

int n,c;

int a=0;

int b=1;

cout<<"Enter the number of terms upto which you want the series\n";

cin>>n;

cout<<a<<" "<<b<<" ";

for(int i=1;i<=n-2;i++){

c=a+b;

cout<<c<<" ";

a=b;

b=c;

}

cout<<endl;

cout<<"Press any key to continue\n";

getch();

return;

}

void calFact(){

int n,fact=1;

cout<<"Enter a number\n";

cin>>n;

for(int i=1;i<=n;i++){

fact\*=i;

}

cout<<"Factorial is = "<<fact;

cout<<endl;

cout<<"Press any key to continue\n";

getch();

return;

}

void calGCD(){

int x,y,gcd;

cout<<"Enter two numbers:\n";

cin>>x>>y;

for(int i=1;i<=x && i<=y;i++){

if(x%i==0 && y%i==0){

gcd=i;

}

}

cout<<"\nGreatest Commmon Divisor is --> "<<gcd;

cout<<endl;

cout<<"Press any key to continue\n";

getch();

}

int main(){

int choice;

do{

cout<<"What do you want?\n";

cout<<"1. Display Fibbonacci series\n";

cout<<"2. Find factorial\n";

cout<<"3. Find GCD\n";

cout<<"4. Exit\n";

cin>>choice;

switch(choice){

case 1:

fibboSeries();

break;

case 2:

calFact();

break;

case 3:

calGCD();

break;

default:

break;

}

}while(choice!=4);

}

Q15.

#include<conio.h>

#include<iostream>

#define MAX\_DIM 100

using namespace std;

class Matrix {

private:

int rows;

int cols;

int matrix[MAX\_DIM][MAX\_DIM];

public:

Matrix(int, int);

void get();

void put();

void sum(Matrix &);

void difference(Matrix &);

void product(Matrix &);

void transpose();

};

Matrix::Matrix(int r = 3, int c = 3) {

rows = r;

cols = c;

for (int i = 0; i < rows; i++){

for (int j = 0; j < cols; j++) {

matrix[i][j] = 0;

}

}

}

void Matrix::get() {

for (int i = 0; i < rows; i++)

for (int j = 0; j < cols; j++) {

cout << "Enter element (" << i << "," << j << "): ";

cin >> matrix[i][j]; } return;

}

void Matrix::put() {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++)

cout << matrix[i][j] << "\t";

cout << endl;

}

}

void Matrix::sum(Matrix &o) {

if (rows != o.rows || cols != o.cols) {

cout << "Matrices cannot be added." << endl; return;

}

Matrix temp(rows, cols);

cout << "SUM:\n\n";

for (int i = 0; i < rows; i++)

for (int j = 0; j < cols; j++)

temp.matrix[i][j] = matrix[i][j] + o.matrix[i][j];

temp.put();

return;

}

void Matrix::difference(Matrix &o) {

if (rows != o.rows || cols != o.cols) {

cout << "Matrices cannot be subtracted." << endl; return;

}

Matrix temp(rows, cols);

cout << "DIFFERENCE:\n\n";

for (int i = 0; i < rows; i++)

for (int j = 0; j < cols; j++)

temp.matrix[i][j] = matrix[i][j] - o.matrix[i][j];

temp.put();

return;

}

void Matrix::product(Matrix &o) {

if (cols != o.rows) {

cout << "Matrices cannot be multiplied." << endl;

return;

}

Matrix temp(rows, o.cols);

cout << "PRODUCT:\n\n";

for (int i = 0; i < rows; i++)

for (int j = 0; j < cols; j++)

for (int k = 0; k < o.rows; k++)

temp.matrix[i][j] += matrix[i][k] \* o.matrix[k][j];

temp.put();

return;

}

void Matrix::transpose() {

Matrix temp(cols, rows);

cout << "TRANSPOSE:\n\n";

for (int i = 0; i < rows; i++)

for (int j = 0; j < cols; j++)

temp.matrix[j][i] = matrix[i][j];

temp.put();

return;

}

void getInputs(Matrix &, Matrix &);

void handleMenu(Matrix &, Matrix &);

int main() {

int r1, c1, r2, c2;

cout << "Enter dimensions of Matrix A: ";

cin >> r1 >> c1;

cout << "Enter dimensions of Matrix B: ";

cin >> r2 >> c2;

Matrix A(r1, c1), B(r2, c2);

getInputs(A, B);

handleMenu(A, B);

cout << "Exiting...\n"; return 0;

}

void getInputs(Matrix &A, Matrix &B) {

cout << "\nMatrix A\n---------\n";

A.get();

cout << endl;

cout << "\nMatrix B\n---------\n";

B.get();

cout << endl;

return;

}

void handleMenu(Matrix &A, Matrix &B) {

int ch = 0;

do {

cout << "\nMenu\n-------\n(1) Add\n(2) Subtract";

cout << "\n(3) Multiply\n(4) Transpose A\n(5) Transpose B";

cout << "\n(6) View Matrices\n(7) Exit\n\nEnter Choice: ";

cin >> ch;

cout << endl;

switch (ch) {

case 1:

A.sum(B);

break;

case 2:

A.difference(B);

break;

case 3:

A.product(B);

break;

case 4:

A.transpose();

break;

case 5:

B.transpose();

break;

case 6:

cout << "Matrix A\n---------\n";

A.put();

cout << endl;

cout << "Matrix B\n---------\n";

B.put();

break;

default:

break;

}

if (ch != 7) {

cout << "\nPress Enter to continue ...\n";

cin.ignore();

cin.get();

}

} while (ch != 7);

return;

}

Q16.

#include <iostream>

#include <cstring>

using namespace std;

class Person {

private:

int age;

char name[255];

public:

Person();

virtual ~Person();

virtual void get();

virtual void put();

};

Person::Person() {

cout << "Constructor of Person called...\n";

strcpy(name, "");

age = 0;

}

Person::~Person() {

cout << "Destructor of Person called...\n";

}

void Person::get() {

cout << "Enter name: ";

cin >> name;

cout << "Enter age: ";

cin >> age; return;

}

void Person::put() {

cout << "Name: " << name << "\n";

cout << "Age: " << age << "\n";

return;

}

class Teacher : public Person {

private:

int facultyId;

char department[255];

public:

Teacher();

~Teacher();

void get();

void put();

};

Teacher::Teacher() {

cout << "Constructor of Teacher called...\n";

facultyId = 0;

strcpy(department, "");

}

Teacher::~Teacher() {

cout << "Destructor of Teacher called...\n";

}

void Teacher::get() {

Person::get();

cout << "Enter faculty ID: ";

cin >> facultyId;

cout << "Enter department: ";

cin >> department;

return;

}

void Teacher::put() {

Person::put();

cout << "Faculty ID: " << facultyId << "\n";

cout << "Department: " << department << "\n";

return;

}

class Student : public Person {

private:

int rollNo;

float marks;

public:

Student();

~Student();

void get();

void put();

};

Student::Student() {

cout << "Constructor of Student called...\n";

rollNo = 0;

marks = 0;

}

Student::~Student() {

cout << "Destructor of Student called...\n";

}

void Student::get() {

Person::get();

cout << "Enter roll no: ";

cin >> rollNo;

cout << "Enter marks: ";

cin >> marks;

return;

}

void Student::put() {

Person::put();

cout << "Roll No: " << rollNo << "\n";

cout << "Marks: " << marks << "\n";

return;

}

int main() {

Person \*person1, \*person2;

cout << "Initializing two instances of Person...\n";

person1 = new Person();

person2 = new Person();

cout << endl;

cout << "Enter details for Person 1: \n";

person1->get();

cout << endl;

cout << "Enter details for Person 2: \n";

person2->get();

cout << endl;

cout << "Person 1\n--------\n";

person1->put();

cout << endl;

cout << "Person 2\n--------\n";

person2->put();

cout << endl;

cout << "Changing Person 1 to Teacher...\n";

person1 = new Teacher();

cout << "Person 1 is now a Teacher...\n";

cout << "Enter new details for Person 1:\n";

person1->get();

cout << endl;

cout << "Changing Person 2 to Student...\n";

person2 = new Student();

cout << "Person 2 is now a Student...\n";

cout << "Enter new details for Person 2:\n";

person2->get();

cout << endl;

cout << "Person 1\n--------\n";

person1->put();

cout << endl;

cout << "Person 2\n--------\n";

person2->put();

cout << endl;

cout << "Deleting Person 1...\n";

delete person1;

cout << endl;

cout << "Deleting Person 2...\n";

delete person2;

cout << endl;

return 0;

}

Q17.

#include <iostream>

using namespace std;

void clrscr() {

#ifdef \_WIN32

system("cls");

#elif \_\_unix\_\_

system("clear");

#endif

}

class Triangle {

private:

int base;

int height;

public:

Triangle(int, int);

void calculateArea();

void calculateArea(float, float);

void operator=(Triangle &);

void operator==(Triangle &);

};

Triangle::Triangle(int x = 0, int y = 0) {

base = x; height = y;

}

void Triangle::calculateArea() {

cout << "Area of Triangle: " << 0.5 \* base \* height << endl;

return;

}

void Triangle::calculateArea(float x, float y) {

cout << "Area (Overloaded): " << 0.5 \* x \* y << endl;

return;

}

void Triangle::operator=(Triangle &o) {

base = o.base;

height = o.height;

return;

}

void Triangle::operator==(Triangle &o) {

if (base == o.base && height == o.height)

cout << "Triangles are Equal." << endl;

else cout << "Triangles are Not Equal." << endl;

return;

}

int main() {

int b, h, ch;

Triangle C;

cout << "Enter base of Triangle 1: ";

cin >> b;

cout << "Enter height of Triangle 1: ";

cin >> h;

Triangle A(b, h);

cout << endl;

cout << "Enter base of Triangle 2: ";

cin >> b;

cout << "Enter height of Triangle 2: ";

cin >> h;

Triangle B(b, h);

do {

clrscr();

cout << "Menu\n-------";

cout << "\n(1) Calculate Area of Triangle 1" << "\n(2) Calculate Area of Triangle 2" << "\n(3) Overload Area Calculation" << "\n(4) Assign A to C" << "\n(5) Check Equality of A and B" << "\n(6) Check Equality of A and C" << "\n(7) Exit\n" << "Enter Choice: ";

cin >> ch;

switch (ch) {

case 1:

A.calculateArea();

break;

case 2:

B.calculateArea();

break;

case 3:

cout << "\nEnter base (for overload): ";

cin >> b;

cout << "Enter height (for overload): ";

cin >> h; A.calculateArea(b, h);

break;

case 4:

C = A;

break;

case 5:

A == B;

break;

case 6:

A == C;

break;

default:

break;

}

if (ch != 7) {

cout << "Press Enter to continue...\n";

cin.ignore();

cin.get();

}

} while (ch != 7);

cout << "\nExiting...\n";

return 0;

}

Q18.

#include <iomanip>

#include <fstream>

#include <iostream>

using namespace std;

int main(int argc, char \*\*argv) {

// ensure proper usage

if (argc != 3) {

cerr << "Usage: ./main infile outfile\n";

return 1;

}

// attempt to open the input stream

ifstream fin(argv[1]);

if (!fin) {

cerr << "Could not read " << argv[1] << "\n";

return 1;

}

// attempt to open the output stream

ofstream fout(argv[2]);

if (!fout) {

cerr << "Could not write to " << argv[1] << "\n";

return 1;

}

// read each character from input stream to

// buffer and write it to the output stream

char ch;

while (fin >> ch){

fout << ch;

}

// close all file i/o streams

fin.close();

fout.close();

return 0;

void Length::display() {

cout << "Length: ";

cout << feet << "\' ";

cout << inch << "\"\n";

return;

}

Length::operator+(Length &o) {

Length temp(\*this);

temp.feet += o.feet;

temp.inch += o.inch;

if (temp.inch >= 12)

while (temp.inch >= 12) {

temp.inch -= 12;

temp.feet++;

}

return temp;

}

Length Length::operator-(Length &o) {

Length temp(\*this);

temp.feet -= o.feet;

temp.inch -= o.inch;

if (temp.inch >= 12)

while (temp.inch >= 12) {

temp.inch -= 12;

temp.feet++;

}

temp.feet = abs(temp.feet);

temp.inch = abs(temp.inch);

return temp;

}

int main() {

Length A, B, sum, diff;

cout << "Length 1\n";

cout << "--------\n";

A.input();

A.display();

cout << endl;

cout << "Length 2\n";

cout << "--------\n";

B.input();

B.display();

cout << endl;

cout << "Sum\n";

cout << "---\n";

sum = A + B;

sum.display();

cout << endl;

cout << "Difference\n";

cout << "----------\n";

diff = A - B;

diff.display();

cout << endl;

return 0;

}

Q19.

#include <iostream>

using namespace std;

class Box {

private:

float length;

float breadth;

float height;

public:

Box();

Box(Box &);

Box(float, float, float);

~Box() {}

void show();

void isCube();

float calculateVolume();

float calculateSurfaceArea();

Box operator+(Box &);

Box operator=(Box &);

friend void operator==(Box &, Box &);

};

Box::Box() {

length = breadth = height = 0;

}

Box::Box(Box &o) {

length = o.length;

breadth = o.breadth;

height = o.height;

}

Box::Box(float a, float b, float c) {

length = a;

breadth = b;

height = c;

}

void Box::show() {

cout << "Length: " << length << " units\n";

cout << "Breadth: " << breadth << " units\n";

cout << "Height: " << height << " units\n";

return;

}

void Box::isCube() {

if (length == breadth && breadth == height && height == length)

cout << "Box is a Cube";

else

cout << "Box is a Cuboid";

cout << endl;

return;

}

float Box::calculateVolume() {

return length \* breadth \* height;

}

float Box::calculateSurfaceArea() {

return 2 \* (length \* breadth + breadth \* height + height \* length);

}

Box Box::operator+(Box &o) {

Box temp(\*this);

temp.length += o.length;

temp.breadth += o.breadth;

temp.height += o.height;

return temp;

}

Box Box::operator=(Box &o) {

length = o.length;

breadth = o.breadth;

height = o.height;

return \*this;

}

void operator==(Box &a, Box &o) {

if (a.length == o.length && a.breadth == o.breadth && a.height == o.height)

cout << "Boxes are Equal";

else

cout << "Boxes are Not Equal";

cout << endl;

return;

}

int main() {

Box C;

int l, b, h;

cout << "Enter dimensions of Box 1: ";

cin >> l >> b >> h;

Box A(l, b, h);

cout << "Enter dimensions of Box 2: ";

cin >> l >> b >> h;

Box B(l, b, h);

cout << endl;

cout << "Box 1\n";

cout << "-----\n";

A.show();

cout << endl;

cout << "Box 2\n";

cout << "-----\n";

B.show();

cout << endl;

cout << "Surface Area\n";

cout << "------------\n";

cout << "Box 1: " << A.calculateSurfaceArea() << " square units" << endl;

cout << "Box 2: " << B.calculateSurfaceArea() << " square units" << endl;

cout << endl;

cout << "Volume\n";

cout << "------\n";

cout << "Box 1: " << A.calculateVolume() << " cubic units" << endl;

cout << "Box 2: " << B.calculateVolume() << " cubic units" << endl;

cout << endl;

cout << "Sum of Box 1 and Box 2\n";

cout << "----------------------\n";

(A + B).show();

cout << endl;

cout << "Assigning Box 1 to Box 3...\n\n";

C = A;

cout << "Equality of Box 1 and Box 2\n";

cout << "---------------------------\n";

A == B;

cout << endl;

cout << "Equality of Box 1 and Box 3:\n";

cout << "---------------------------\n";

A == C;

cout << endl;

return 0;

}

Q20.

#include <iostream>

#include <cstdlib>

using namespace std;

class Length {

private:

int feet;

int inch;

public:

Length();

Length(Length &);

~Length() {}

void input();

void display();

Length operator+(Length &);

Length operator-(Length &);

};

Length::Length() {

feet = inch = 0;

}

Length::Length(Length &o) {

feet = o.feet;

inch = o.inch;

}

void Length::input() {

cout << "Enter feet: ";

cin >> feet;

cout << "Enter inch: ";

cin >> inch;

if (inch >= 12) while (inch >= 12) {

inch -= 12;

feet++;

}

return;

}