**Phase-5 Functional Programs**

**Practical-1**

**Aim: Create a Calculator in C++ by using all types of user**

**defined functions. User can perform all types of basic**

**arithmetic operations until he/she wants.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

void menu()

{

cout<<" [1] Addition "<<endl;

cout<<" [2] Substraction "<<endl;

cout<<" [3] Multiplication "<<endl;

cout<<" [4] Divition "<<endl;

cout<<" [5] Modulas "<<endl;

cout<<" [0] Exit "<<endl;

}

void Addition()

{

int n1, n2;

cout<<" Enter Value of A: ";

cin>>n1;

cout<<" Enter Value of B: ";

cin>>n2;

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

cout<<" Addition of above number is "<<n1+n2<<endl;

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

}

void Substraction()

{

int n1, n2;

cout<<" Enter Value of A: ";

cin>>n1;

cout<<" Enter Value of B: ";

cin>>n2;

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

cout<<" Substraction of above number is "<<n1-n2<<endl;

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

}

void Multiplication()

{

int n1, n2;

cout<<" Enter Value of A: ";

cin>>n1;

cout<<" Enter Value of B: ";

cin>>n2;

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

cout<<" Multiplication of above number is "<<n1\*n2<<endl;

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

}

void Divition()

{

int n1, n2;

cout<<" Enter Value of A: ";

cin>>n1;

cout<<" Enter Value of B: ";

cin>>n2;

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

cout<<" Divition of above number is "<<n1/n2<<endl;

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

}

void Modulas()

{

int n1, n2;

cout<<" Enter Value of A: ";

cin>>n1;

cout<<" Enter Value of B: ";

cin>>n2;

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

cout<<" Modulas of above number is "<<n1%n2<<endl;

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

}

class Calculator

{

public:

int c;

void operation()

{

do{

menu();

cout<<" Enter Your Choice: ";

cin>>c;

cout<<endl;

if(c==1)

{

Addition();

}

else if(c==2)

{

Substraction();

}

else if(c==3)

{

Multiplication();

}

else if(c==4)

{

Divition();

}

else if(c==5)

{

Modulas();

}

else if(c!=0)

{

cout<<" Enter Valid Value..."<<endl;

}

}while(c!=0);

}

};

int main()

{

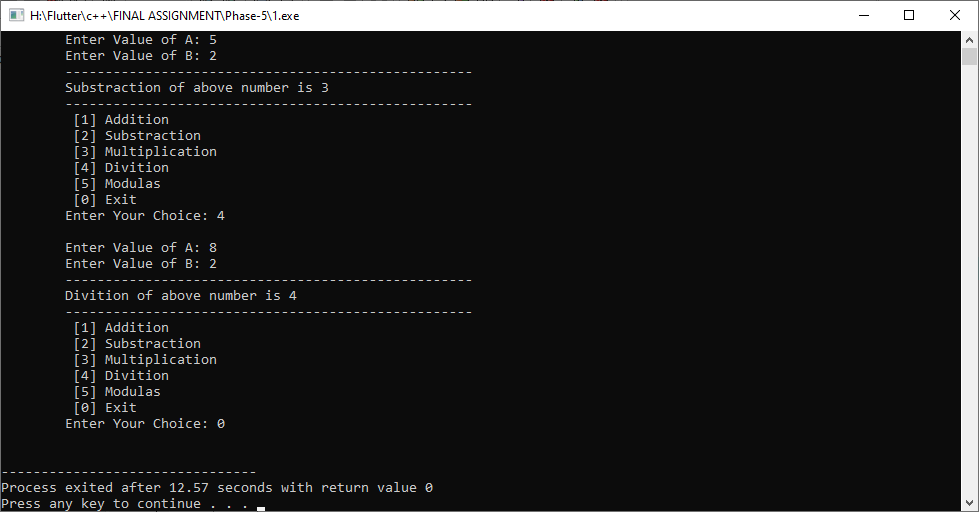
Calculator c1;

c1.operation();

return 0;

}

**Output:**

****

**Practical-2**

**Aim: Develop a solution for Akshay by which he can retrieve factorial of all numbers between given range of two numbers using a C++ user defined function (UDF).**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

void factorial()

{

int n1, n2, fact=1, i;

cout<<endl<<" Enter Starting Number: ";

cin>>n1;

cout<<" Enter Ending Number: ";

cin>>n2;

if(n1<n2)

{

for(i=n1;i<=n2;i++)

{

fact\*=i;

}

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

cout<<" Factorial of "<<n1<<" to "<<n2<<" is: "<<fact<<endl;

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

}

else

{

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

cout<<" Please, Enter Starting Number Less Than Ending Number"<<endl;

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

}

}

class Akshay

{

public:

void fact()

{

factorial();

}

};

int main()

{

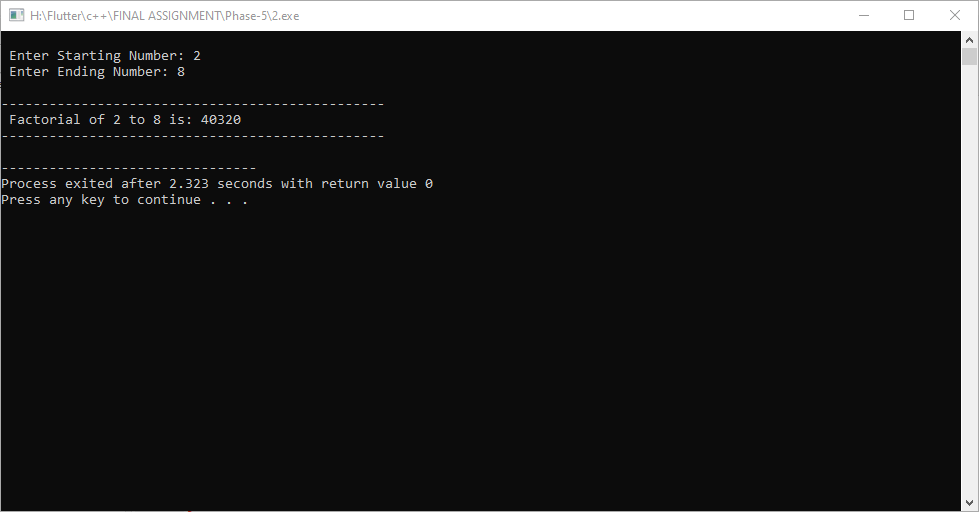
Akshay a1;

a1.fact();

return 0;

}

**Output:**

****

**Practical-3**

**Aim: Kevin has two plain floors within different bowls containing one coin in each bowl. He bet his friend to transfer that coins in either bowls within 5 minutes. Help him by providing a C++ solution using UDF.**

**Program:**

#include<iostream>

using namespace std;

void Bowl()

{

int a[10], b[10];

int i;

cout<<endl<<" Enter 10 Value: "<<endl;

cout<<"================================================="<<endl;

for(i=0;i<10;i++)

{

cout<<" a["<<i<<"] : ";

cin>>a[i];

}

cout<<endl<<" Transfering in Another one : "<<endl;

cout<<"================================================="<<endl;

for(i=0;i<10;i++)

{

b[i]=a[i];

cout<<" b["<<i<<"] : "<<b[i]<<endl;

}

}

class Kevin

{

public:

void solution()

{

Bowl();

}

};

int main()

{

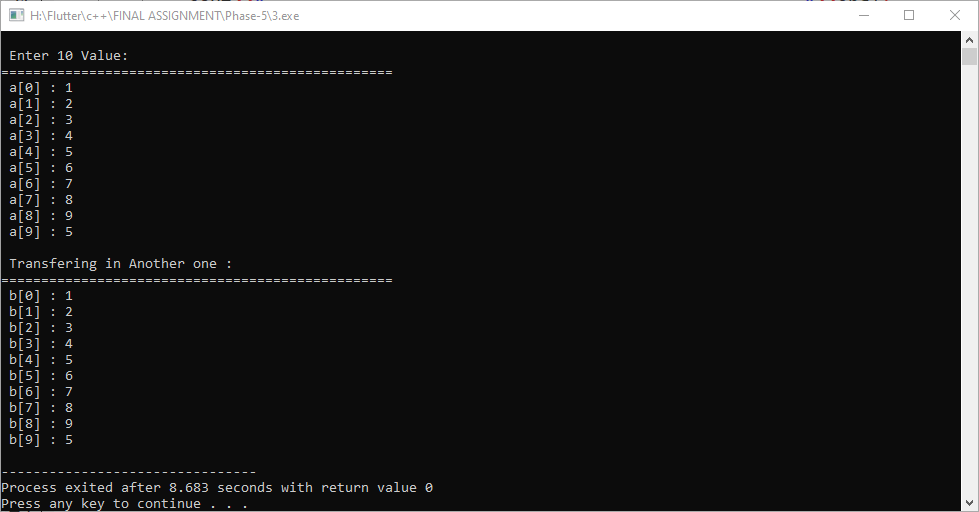
Kevin k1;

k1.solution();

return 0;

}

**Output:**



**PRACTICAL - 4**

**AIM**: **Design a C++ UDF which producing cubes of all elements**

**of provided array in form of another array. ¶hen, find**

**average value of that new array. Based on that average**

**value decide that array’s kindµ**

**If 22<=average<=35, then an array is “¶IGH¶Eª©**

**If 35<average<=50, then an array is “BALANCED©**

**If average>50, then an array is “¶OXIC©**

**If average<22, then an array is “LOOSEª”**

**PROGRAM**:

//4. Design a C++ UDF which producing cubes of all elements

//of provided array in form of another array. ¶hen, find

//average value of that new array. Based on that average

//value decide that array’s kindµ

// If 22<=average<=35, then an array is “¶IGH¶Ea©

// If 35<average<=50, then an array is “BALANCED©

// If average>50, then an array is “¶OXIC©

// If average<22, then an array is “LOOSEa”

#include<iostream>

using namespace std;

void setdata()

{

int a[100],b[100],sum;

int n,i,c,ave=0;

cout<<endl<<" Enter How Many Elements: ";

cin>>n;

cout<<endl;

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

{

cout<<"a["<<i<<"]: ";

cin>>a[i];

}

cout<<endl<<" Cube of All Elements: ";

cout<<"===================================="<<endl;

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

{

sum=a[i]\*a[i]\*a[i];

ave+=a[i]\*a[i]\*a[i];

b[i]=sum;

}

c=ave/n;

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

{

cout<<"b["<<i<<"]: ";

cout<<b[i]<<endl;

}

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

cout<<" Average of All Cube Numbers is: "<<c<<endl;

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

cout<<endl<<" Array Kind is:"<<endl;

cout<<"===================================="<<endl;

if(c>22&&c<=35)

{

cout<<" Tighter"<<endl;

}

else if(c>35&&c<=50)

{

cout<<" Balanced"<<endl;

}

else if(c>50)

{

cout<<" Toxic"<<endl;

}

else if(c<22)

{

cout<<" Looser"<<endl;

}

}

class Udf

{

public:

void Getdata()

{

setdata() ;

}

};

int main()

{

Udf u1;

u1.Getdata();

return 0;

}

**PRACTICAL - 6**

**AIM**: **A Reality show on TV organizes “Fastest-fingers Fast” round for entering in a Game. In this round participant has to find reverse of a given number as soon as possible to win this round. Design a C++ UDF for that.**

**PROGRAM**:

//6. A Reality show on TV organizes “Fastest-fingers Fast”

//round for entering in a Game. In this round participant

//has to find reverse of a given number as soon as possible

//to win this round. Design a C++ UDF for that.

#include<iostream>

using namespace std;

void reverse()

{

int n,d,sum=0;

cout<<endl<<" Enter Number: ";

cin>>n;

while(n!=0)

{

d=n%10;

sum=(sum\*10)+d;

n=n/10;

}

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

cout<<" Reverse of given Number is : "<<sum<<endl;

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

}

class Fastest\_fingers\_Fast

{

public:

void win()

{

reverse();

}

};

int main()

{

Fastest\_fingers\_Fast f1;

f1.win();

return 0;

}

**PRACTICAL - 7**

**AIM**: **Ajay has to find Fibonacci Series upto given number to successfully pass in Math’s examination. Help him by designing a UDF in C++.**

**PROGRAM**:

//7. Ajay has to find Fibonacci Series upto given number to

//successfully pass in Math’s examination. Help him by

//designing a UDF in C++.

#include<iostream>

using namespace std;

void f\_s()

{

int n,d,sum=0,i,n1=0,n2=1,n3;

cout<<endl<<" Enter Number: ";

cin>>n;

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

cout<<" Fibonacci Series is : 0, 1, ";

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

{

n3=n1+n2;

cout<<n3<<" , ";

n1=n2;

n2=n3;

}

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

}

class Fibonacci\_Series

{

public:

void number()

{

f\_s();

}

};

int main()

{

Fibonacci\_Series f1;

f1.number();

return 0;

}

**PRACTICAL - 8**

**AIM**: **Design a C++ UDF which converts given seconds into time in format of HH:MM:SS. Also create another UDF which converts given time into total seconds. End user have choice to perform either operations whenever he/she wants.**

**PROGRAM**:

//8. Design a C++ UDF which converts given seconds into

//time in format of HH:MM:SS. Also create another UDF which

//converts given time into total seconds. End user have

//choice to perform either operations whenever he/she

//wants.

#include<iostream>

using namespace std;

void Menu()

{

cout<<endl<<" >>>>>>>>>>>>>>>MENU<<<<<<<<<<<<<<<"<<endl;

cout<<" =================================="<<endl;

cout<<" || [1] Seconds into Time ||"<<endl;

cout<<" || [2] Time into Seconds ||"<<endl;

cout<<" || [0] Exit ||"<<endl;

cout<<" =================================="<<endl<<endl;

}

void Seconds\_into\_Time()

{

int n,a,h,m,s;

cout<<endl<<" Enter Total Seconds: ";

cin>>n;

h=n/3600;

a=n%3600;

m=a/60;

s=a%60;

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

cout<<" Total Time is: "<<h<<":"<<m<<":"<<s<<endl;

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

}

void Time\_into\_Seconds()

{

int n,a,h,m,s,t;

cout<<endl<<" Enter Hours: ";

cin>>h;

cout<<" Enter Minutes: ";

cin>>m;

cout<<" Enter Seconds: ";

cin>>s;

t=(h\*3600)+(m\*60)+s;

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

cout<<" Total Second is: "<<t<<endl;

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

}

class operation

{

public:

int c;

void time()

{

do{

Menu();

cout<<" Enter Your Choice: ";

cin>>c;

cout<<endl;

if(c==1)

{

Seconds\_into\_Time();

}

else if(c==2)

{

Time\_into\_Seconds();

}

else if(c!=0)

{

cout<<" Enter Valid Value..."<<endl;

}

}while(c!=0);

}

};

int main()

{

operation o1;

o1.time();

return 0;

}

**PRACTICAL - 9**

**AIM**: **A Supreme Court wants a system which automatically figure out difference of two given time whether it is in seconds or any other format. Develop a solution in C++ using UDF.**

**PROGRAM**:

//9. A Supreme Court wants a system which automatically

//figure out difference of two given time whether it is in

//seconds or any other format. Develop a solution in C++

//using UDF.

#include<iostream>

using namespace std;

void menu()

{

cout<<endl<<" >>>>>>>>>>>MENU<<<<<<<<<<<"<<endl;

cout<<" =========================="<<endl;

cout<<" || [1] Seconds ||"<<endl;

cout<<" || [2] Hours ||"<<endl;

cout<<" || [3] Minutes ||"<<endl;

cout<<" || [4] Time ||"<<endl;

cout<<" || [0] Exit ||"<<endl;

cout<<" =========================="<<endl<<endl;

}

void Seconds()

{

int s1, s2;

cout<<endl<<" Enter First Seconds: ";

cin>>s1;

cout<<endl<<" Enter Last Seconds: ";

cin>>s2;

if(s1<s2)

{

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

cout<<" Difference of two given time is: "<<s2-s1<<endl;

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

}

else if(s1>s2)

{

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

cout<<" Difference of two given time is: "<<s1-s2<<endl;

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

}

}

void Hours()

{

int h1, h2;

cout<<endl<<" Enter First Hours: ";

cin>>h1;

cout<<endl<<" Enter Last Hours: ";

cin>>h2;

if(h1<h2)

{

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

cout<<" Difference of two given time is: "<<h2-h1<<endl;

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

}

else if(h1>h2)

{

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

cout<<" Difference of two given time is: "<<h1-h2<<endl;

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

}

}

void Minutes()

{

int m1, m2;

cout<<endl<<" Enter First Minutes: ";

cin>>m1;

cout<<endl<<" Enter Last Minutes: ";

cin>>m2;

if(m1<m2)

{

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

cout<<" Difference of two given time is: "<<m2-m1<<endl;

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

}

else if(m1>m2)

{

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

cout<<" Difference of two given time is: "<<m1-m2<<endl;

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

}

}

void Time()

{

int h1,m1,s1,h2,m2,s2,a,b,c,e,h3,m3,s3;

cout<<endl<<" Enter Hours: ";

cin>>h1;

cout<<" Enter Minutes: ";

cin>>m1;

cout<<" Enter Seconds: ";

cin>>s1;

cout<<endl<<" Enter Hours: ";

cin>>h2;

cout<<" Enter Minutes: ";

cin>>m2;

cout<<" Enter Seconds: ";

cin>>s2;

if(h1>h2)

{

a=s1-s2;

s3=a%60;

b=a/60;

c=m1-m2-b;

m3=c%60;

e=c/60;

h3=h1-h2-e;

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

cout<<"Substraction of above time is "

<<h3<<" hours "

<<m3<<" minutes "

<<s3<<" seconds "<<endl;

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

}

else if(h1<h2)

{

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

cout<<" Please, Enter First Time Greater Than Second Time..."<<endl;

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

}

}

class time

{

public:

int c;

void t()

{

do{

menu();

cout<<" Enter Your Choice: ";

cin>>c;

cout<<endl;

if(c==1)

{

Seconds();

}

else if(c==2)

{

Hours();

}

else if(c==3)

{

Minutes();

}

else if(c==4)

{

Time();

}

else if(c!=0)

{

cout<<" Enter Valid Value..."<<endl;

}

}while(c!=0);

}

};

int main()

{

time t1;

t1.t();

return 0;

}

**PRACTICAL - 10**

**AIM**: **A bomb is planted at Suratgarh Railway Station. It can be defused by entering any number which is itself an Armstrong number. Design a C++ UDF which figures out if a given number is Armstrong or not.**

**PROGRAM**:

//10. A bomb is planted at Suratgarh Railway Station. It

//can be defused by entering any number which is itself an

//Armstrong number. Design a C++ UDF which figures out if a

//given number is Armstrong or not.

#include<iostream>

using namespace std;

void Armstrong\_number()

{

int n,d,i,count=0,j,sum,m;

int a[100];

cout<<endl<<" Enter Number: ";

cin>>n;

m=n;

while(n!=0)

{

d=n%10;

a[i]=d;

i++;

n=n/10;

count++;

}

for(j=0;j<count;j++)

{

// cout<<a[j]<<endl;

sum = sum + ( a[j]\*a[j]\*a[j] );

}

if(m==sum)

{

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

cout<<" Yes, Number is Armstrong."<<endl;

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

}

else

{

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

cout<<" No, Number is not Armstrong."<<endl;

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

}

}

class Number

{

public:

void n()

{

Armstrong\_number();

}

};

int main()

{

Number a1;

a1.n();

return 0;

**PRACTICAL - 11**

**AIM**: **Declare a result of the survey that tells us which country have largest Army strength, US, China or India. Design a C++ UDF to announce the result of this survey to the public.**

**PROGRAM**:

//11. Declare a result of the survey that tells us which

//country have largest Army strength, US, China or India.

//Design a C++ UDF to announce the result of this survey to

//the public.

#include<iostream>

using namespace std;

void result()

{

int india, US, china;

india=1237117;

US=480893;

china=915000;

if(india>US)

{

if(india>china)

{

cout<<endl<<" India have largest Army strength."<<endl;

}

else

{

cout<<endl<<" China have largest Army strength."<<endl;

}

}

else

{

if(US>china)

{

cout<<endl<<" US have largest Army strength."<<endl;

}

else

{

cout<<endl<<" China have largest Army strength."<<endl;

}

}

}

class country

{

public:

void getdata()

{

result();

}

};

int main()

{

country c1;

c1.getdata();

return 0;

}

**PRACTICAL - 12**

**AIM**: **Two buses(Bus B1 & Bus B2) head forwards from Mumbai to Kolkata. Both of them have to cover total distance of 1933 KM. Bus B1 reached on destination with total time of 40 Hr & Bus B2 takes total time of 46 Hr. Find out velocity of both buses using a C++ UDF.**

**PROGRAM**:

//12. Two buses(Bus B1 & Bus B2) head forwards from Mumbai

//to Kolkata. Both of them have to cover total distance of

//1933 KM. Bus B1 reached on destination with total time of

//40 Hr & Bus B2 takes total time of 46 Hr. Find out

//velocity of both buses using a C++ UDF.

#include<iostream>

using namespace std;

void velocity()

{

int b1,b2;

int d=1933,t1=40,t2=46;

b1=d/t1;

b2=d/t2;

cout<<endl<<" Velocity of"<<endl;

cout<<"================================================"<<endl;

cout<<" Bus b1 = "<<b1<<endl;

cout<<" Bus b2 = "<<b2<<endl;

}

class Bus

{

public:

void getdata()

{

velocity();

}

};

int main()

{

Bus b1;

b1.getdata();

return 0;

}

**PRACTICAL - 13**

**AIM**: **Develop a C++ solution for Maths students to solve**

**all types Geometry problems such likeI**

**L Area of Circlg**

**L Perimeter of Circlg**

**L Area of Squarg**

**L Area of Rectanglg**

**L Area of Trianglg**

**L Area of Sphere**

**PROGRAM**:

//13. Develop a C++ solution for Maths students to solve

//all types Geometry problems such likeI

//L Area of Circlg

//L Perimeter of Circlg

//L Area of Squarg

//L Area of Rectanglg

//L Area of Trianglg

//L Area of Sphere

#include<iostream>

using namespace std;

void menu()

{

cout<<endl<<" >>>>>>>>>>>MENU<<<<<<<<<<<"<<endl;

cout<<" =============================="<<endl;

cout<<" || [1] Area of Circle ||"<<endl;

cout<<" || [2] Perimeter of Circle ||"<<endl;

cout<<" || [3] Area of Square ||"<<endl;

cout<<" || [4] Area of Rectangle ||"<<endl;

cout<<" || [5] Area of Triangle ||"<<endl;

cout<<" || [6] Area of Sphere ||"<<endl;

cout<<" || [0] Exit ||"<<endl;

cout<<" =============================="<<endl<<endl;

}

void Area\_of\_Circle()

{

int r;

const float PI=3.14;

float c;

cout<<endl<<" Enter Radious: ";

cin>>r;

c=PI\*(r\*r);

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

cout<<" Area of Circle is: "<<c<<endl;

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

}

void Perimeter\_of\_Circle()

{

int r;

const float PI=3.14;

float c;

cout<<endl<<" Enter Radious: ";

cin>>r;

c=2\*(PI\*r);

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

cout<<" Perimeter of Circle is: "<<c<<endl;

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

}

void Area\_of\_Square()

{

int l;

int s;

cout<<endl<<" Enter Length: ";

cin>>l;

s=l\*l;

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

cout<<" Area of Square is: "<<s<<endl;

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

}

void Area\_of\_Rectangle()

{

int l,w;

int r;

cout<<endl<<" Enter Length: ";

cin>>l;

cout<<endl<<" Enter Width: ";

cin>>w;

r=l\*w;

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

cout<<" Area of Rectangle is: "<<r<<endl;

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

}

void Area\_of\_Triangle()

{

int b,h;

float t;

cout<<endl<<" Enter Base: ";

cin>>b;

cout<<endl<<" Enter Heigth: ";

cin>>h;

t=0.5\*(b\*h);

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

cout<<" Area of Triangle is: "<<t<<endl;

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

}

void Area\_of\_Sphere()

{

int r;

const float PI=3.14;

float s;

cout<<endl<<" Enter Radious: ";

cin>>r;

s=4\*PI\*(r\*r);

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

cout<<" Area of Sphere is: "<<s<<endl;

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

}

class Geometry\_problems

{

public:

int c;

void solution()

{

do{

menu();

cout<<" Enter Your Choice: ";

cin>>c;

cout<<endl;

if(c==1)

{

Area\_of\_Circle();

}

else if(c==2)

{

Perimeter\_of\_Circle();

}

else if(c==3)

{

Area\_of\_Square();

}

else if(c==4)

{

Area\_of\_Rectangle();

}

else if(c==5)

{

Area\_of\_Triangle();

}

else if(c==6)

{

Area\_of\_Sphere();

}

else if(c!=0)

{

cout<<" Enter Valid Value..."<<endl;

}

}while(c!=0);

}

};

int main()

{

Geometry\_problems g1;

g1.solution();

return 0;

}

**PRACTICAL - 14**

**AIM**: **A window on a side wall have a dimension of 10x4 feet. Kaveri wants to apply curtains on that window such that a window will perfectly coverd from all sides with extra 2 feet. Design a C++ UDF with figures out if a given dimensions of curtains satisfies mentioned criteria or not.**

**PROGRAM**:

//14. A window on a side wall have a dimension of 10x4

//feet. Kaveri wants to apply curtains on that window such

//that a window will perfectly coverd from all sides with

//extra 2 feet. Design a C++ UDF with figures out if a

//given dimensions of curtains satisfies mentioned criteria

//or not.

#include<iostream>

#include<stdio.h>

#include<string.h>

using namespace std;

void Kaveri()

{

int l=10, w=4;

int c;

c=(l\*w)+8;

cout<<endl<<" A window on a side wall have a dimension of 10x4 feet."<<endl;

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

cout<<" For Covering Window is: "<<endl;

cout<<" Kaveri needs "<<c<<" area curtain."<<endl;

}

class curtain

{

public:

void getdata()

{

Kaveri();

}

};

int main()

{

curtain c1;

c1.getdata();

return 0;

}