Experiment – 2

Aim: Write a function to raise a number m to power n. The function takes a double value for m and integer value for n. Use default value for n to make the function to calculate squares

**Source Code:**

#include<iostream>

using namespace std;

class Power{

public:

double m;

int n;

double ans=1;

void input(){

cout<<"Enter number: ";

cin>>m;

cout<<"Enter power: ";

cin>>n;

}

void Pow(){

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

ans = ans \* m;

}

}

void display(){

cout<<"Result of "<<m<<" raise to power "<<n<<" = "<<ans<<endl;

}

};

int main(){

Power obj;

obj.input();

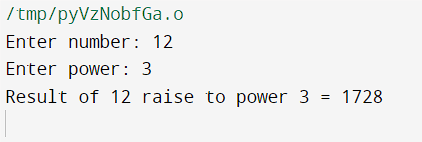
obj.Pow();

obj.display();

return 0;

}

**OUTPUT:**



Experiment – 3

Aim- Calculate area of circle, rectangle and triangle by method overloading.

**Source Code:**

#include <iostream>

#include <math.h>

using namespace std;

class areaCalc{

public:

void area(float r){

cout << "\n Area of Circle= " << 3.14 \* r \* r;

}

void area(int l, int b){

cout << "\n Area of Rectangle= " << l \* b ;

}

void area(float b, floath){

int ar

ar = 0.5\*b\*h;

cout << "\n Area of Triangle= " << ar ;

}

};

int main(){

areaCalc obj;

int r, l, b, a, c;

cout << " 1- Rectangle\n 2- Circle\n 3- Triangle";

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

cin >> c;

switch (c)

{

case 1:

cout << "\n Length and Breadth of rectangle: ";

cin >> l >> b;

obj.area(l, b);

break;

case 2:

cout << "\n Radius of circle: ";

cin >> r;

obj.area(r);

break;

case 3:

cout << "\n Two sides of triangle: ";

cin >> a >> b;

obj.area(a, b);

break;

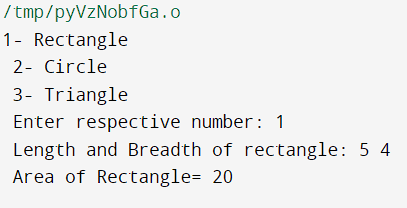
default:

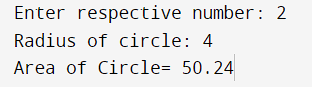
cout << "invalid choice";

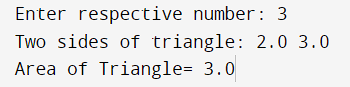
}

}

**OUTPUT:**







Experiment – 4

Aim: Create a class time with members hours, minutes and seconds. Take input, add two time objects and pass objects as argument to function and display result.

**Source Code:**

#include <iostream> using namespace std;

class Time

{

int hours, minutes, seconds;

public:

void setTime();

void addTime(Time, Time);

};

void Time :: setTime()

{

cout << "Enter hours, minutes and seconds: " << endl; cin >> hours >> minutes >> seconds;

}

void Time :: addTime(Time t1, Time t2)

{

int hr, m, s;

s = t1.seconds + t2.seconds;

m = t1.minutes + t2.minutes + s / 60; hr = t1.hours + t2.hours + m / 60;

s %= 60;

m %= 60;

cout << "The time after addition is: " << endl

<< hr << ":" << m << ":" << s;

}

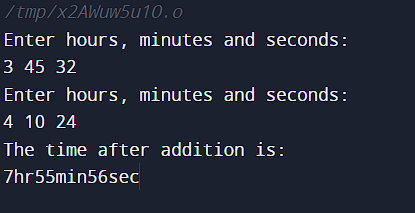
int main()

{

Time t1, t2, t3; t1.setTime();

t2.setTime(); t3.addTime(t1, t2); return 0;

}

**OUTPUT:**