Section : B

Name : Muhammad usama.

Roll no : 17f-8195.

Assignment # 5

Task 1:

#include<iostream>

using namespace std;

class time

{

int mints;

int second;

int hours;

public:

time()

{

mints=0;

second=0;

hours=0;

}

time operator +(time obj)

{

time temp;

temp.mints=mints+obj.mints;

if(temp.mints>=60)

{

temp.mints=0;

temp.hours=temp.hours+1;

}

temp.second=second+obj.second;

if(temp.second>=60)

{

temp.second=0;

temp.mints++;

}

temp.hours=hours+obj.hours;

return temp;

}

int operator >(time obj)

{

if(obj.hours==hours)

{

if(obj.mints>mints)

{

return 1;

}

if(obj.mints==mints)

{

if(obj.second>second)

{

return 1;

}

else

{

return 0;

}

}

}

if(obj.hours>hours)

{

return 1;

}

else

{

return 0;

}

}

int operator <(time obj)

{

if(obj.hours==hours)

{

if(obj.mints<mints)

{

return 0;

}

if(obj.mints==mints)

{

if(obj.second<second)

{

return 0;

}

}

}

if(obj.hours<hours)

{

return 0;

}

else

{

return 0;

}

}

friend istream &operator >>(istream &in,time &obj);

friend ostream &operator <<(ostream &out,time &obj);

};

istream &operator >>(istream &in,time &obj)

{

cout<<"enter second : ";

in>>obj.second;

cout<<"enter mints : ";

in>>obj.mints;

cout<<"eneter hours : ";

in>>obj.hours;

return in;

}

ostream &operator <<(ostream &out,time &obj)

{

out<<obj.hours<<" : "<<obj.mints<<" : "<<obj.second<<endl;

return out;

}

int main()

{

time obj1,obj2;

cout<<"enetr for obj1"<<endl;

cin>>obj1;

cout<<"eneter for obj2"<<endl;

cin>>obj2;

cout<<"sum of time is "<< obj1+obj2;

if(obj1>obj2==1)

{

cout<<"obj2 is greater"<<endl;

}

else

{

cout<<"obj1 is greater"<<endl;

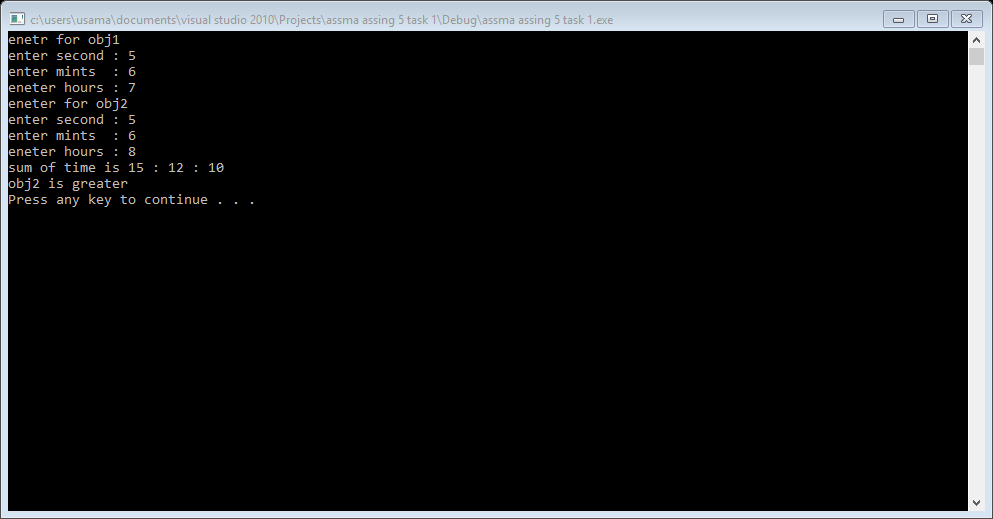
}

system("pause");

return 0;

}

Result:



Task 2:

#include<iostream>

using namespace std;

class circle

{

int radius;

public:

circle(int r)

{

radius=r;

}

void show()

{

cout<<"radius : "<<radius<<endl;

}

circle()

{

}

circle operator +(circle obj)

{

circle temp;

temp.radius=radius+obj.radius;

return temp;

}

circle operator -(circle obj)

{

circle temp;

temp.radius=radius-obj.radius;

return temp;

}

circle operator +=(circle obj)

{

circle temp;

temp.radius=radius+obj.radius;

return temp;

}

int operator <(circle obj)

{

if(obj.radius>radius)

{

return 1;

}

if(obj.radius<radius)

{

return 0;

}

}

int operator >(circle obj)

{

if(obj.radius<radius)

{

return 0;

}

if(obj.radius>radius)

{

return 1;

}

}

int operator ==(circle obj)

{

if(obj.radius==radius)

{

return 1;

}

else

{

return 0;

}

}

int operator !=(circle obj)

{

if(obj.radius!=radius)

{

return 1;

}

else

{

return 0;

}

}

};

int main()

{

circle obj1,obj2,obj3;

int r;

cout<<"enetr radius for first obj : ";

cin>>r;

obj2=r;

cout<<"enter radius for 2nd obj : ";

cin>>r;

obj1=r;

obj3=obj1+obj2;

cout<<"sum of two circle"<<endl;

obj3.show();

obj3=obj1-obj2;

cout<<"subtraction of 2 circle"<<endl;

obj3.show();

if(obj1<obj2)

{

cout<<"radius of obj2 is greater"<<endl;

}

else

{

cout<<"radius of obj1 is greater"<<endl;

}

(obj1+=5).show();

if(obj1==obj2)

{

cout<<"objects are equal "<<endl;

}

else

{

cout<<"objects are not equal "<<endl;

}

if(obj1!=obj2)

{

cout<<"objects are not equal"<<endl;

}

else

{

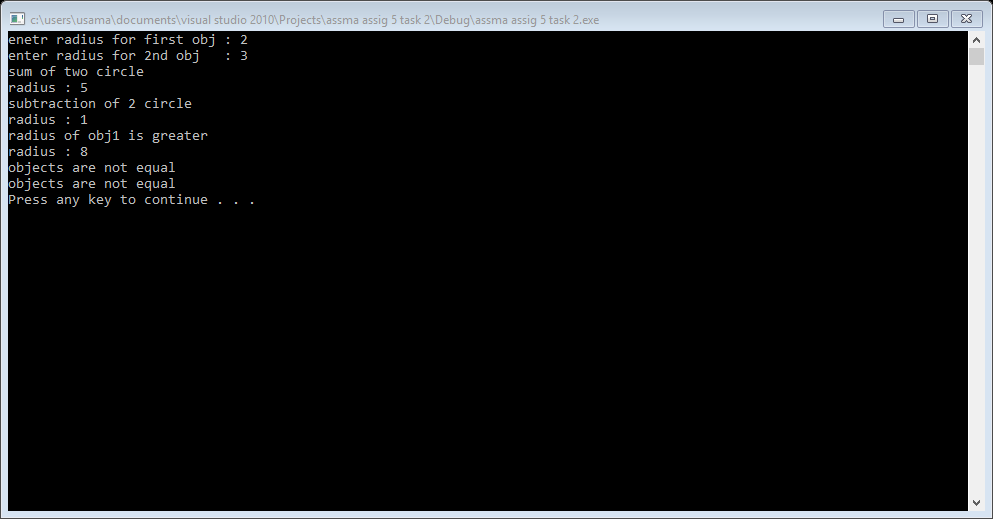
cout<<"objects are not equal "<<endl;

}

system("pause");

return 0;}

Result:



Task 4:

#include<iostream>

using namespace std;

class polynomial

{

int x;

int y;

public:

polynomial(int v):x(v)

{

y=2\*(x\*x\*x\*x);

}

polynomial()

{

}

void sety(int v)

{

y=v;

}

void setx(int v)

{

x=v;

}

void show()

{

cout<<"y : "<<y<<endl;

}

polynomial operator +(polynomial obj)

{

polynomial temp;

temp.y=y+obj.y;

return temp;

}

polynomial operator -(polynomial obj)

{

polynomial temp;

temp.y=y-obj.y;

return temp;

}

polynomial operator \*(polynomial obj)

{

polynomial temp;

temp.y=y\*obj.y;

return temp;

}

polynomial operator +=(polynomial obj)

{

polynomial temp;

temp.y=y+obj.y;

return temp;

}

polynomial operator -=(polynomial obj)

{

polynomial temp;

temp.y=y-obj.y;

return temp;

}

polynomial operator \*=(polynomial obj)

{

polynomial temp;

temp.y=y\*obj.y;

return temp;

}

bool operator ==(polynomial obj)

{

if(obj.y==y)

{

return 1;

}

else

{

return 0;

}

}

bool operator !=(polynomial obj)

{

if(obj.y!=y)

{

return 1;

}

else

{

return 0;

}

}

};

void addition();

void subtraction();

void multiplication();

void equality();

void unequality();

int main()

{

int c=0;

do

{

cout<<"1 : for addition"<<endl;

cout<<"2 :for subtraction"<<endl;

cout<<"3 :for multiplication"<<endl;

cout<<"4 : for equality simble "<<endl;

cout<<"5 : for unequality simble"<<endl;

cout<<"6 : for exit "<<endl;

cin>>c;

system("cls");

if(c==1)

{

addition();

}

if(c==2)

{

subtraction();

}

if(c==3)

{

multiplication();

}

if(c==4)

{

equality();

}

if(c==5)

{

unequality();

}

}

while(c!=6);

system("pause");

return 0;

}

void addition()

{

int x=0;

cout<<"eneter value of x for 1st obj : ";

cin>>x;

polynomial obj1(x);

cout<<"eneter value of x for 2nd obj : ";

cin>>x;

polynomial obj2(x);

cout<<"sum is : ";

(obj1+obj2).show();

cout<<"sum b by 2nd method"<<endl;

cout<<"sum : ";

(obj1+=obj2).show();

}

void subtraction()

{

int x=0;

cout<<"eneter value of x for 1st obj : ";

cin>>x;

polynomial obj1(x);

cout<<"eneter value of x for 2nd obj : ";

cin>>x;

polynomial obj2(x);

cout<<"subtraction is : ";

(obj1-obj2).show();

cout<<endl;

cout<<"subtraction by 2nd method"<<endl;

cout<<"subtraction : ";

(obj1-=obj2).show();

}

void multiplication()

{

int x=0;

cout<<"eneter value of x for 1st obj : ";

cin>>x;

polynomial obj1(x);

cout<<"eneter value of x for 2nd obj : ";

cin>>x;

polynomial obj2(x);

cout<<"mutiplication is : ";

(obj1\*obj2).show();

cout<<endl;

cout<<"multiplication by 2nd method"<<endl;

cout<<"multiplication : ";

(obj1\*=obj2).show();

}

void equality()

{

int x=0;

cout<<"eneter value of x for 1st obj : ";

cin>>x;

polynomial obj1(x);

cout<<"eneter value of x for 2nd obj : ";

cin>>x;

polynomial obj2(x);

if(obj1==obj2)

{

cout<<"objects are equal"<<endl;

}

else

{

cout<<"objects are not equal"<<endl;

}

}

void unequality()

{

int x=0;

cout<<"eneter value of x for 1st obj : ";

cin>>x;

polynomial obj1(x);

cout<<"eneter value of x for 2nd obj : ";

cin>>x;

polynomial obj2(x);

if(obj1!=obj2)

{

cout<<"objects are not equal"<<endl;

}

else

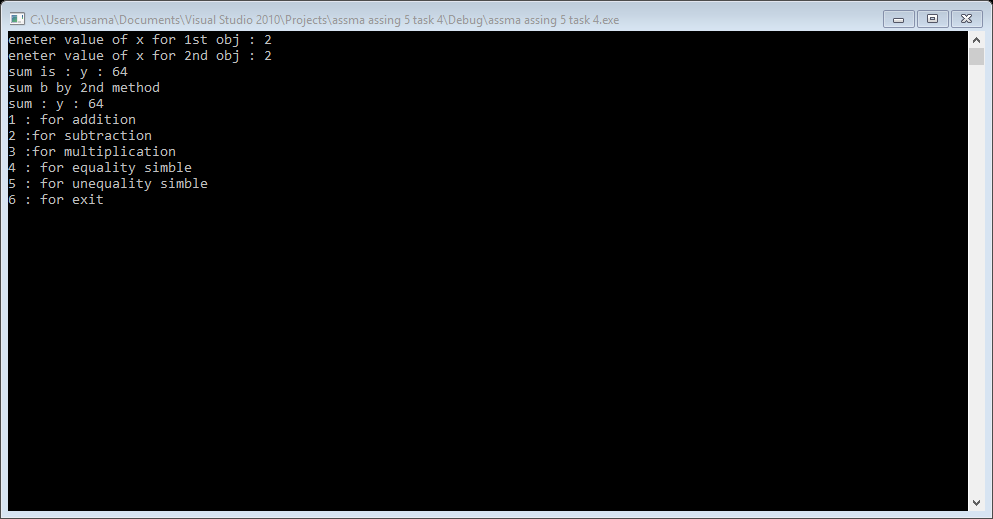
{

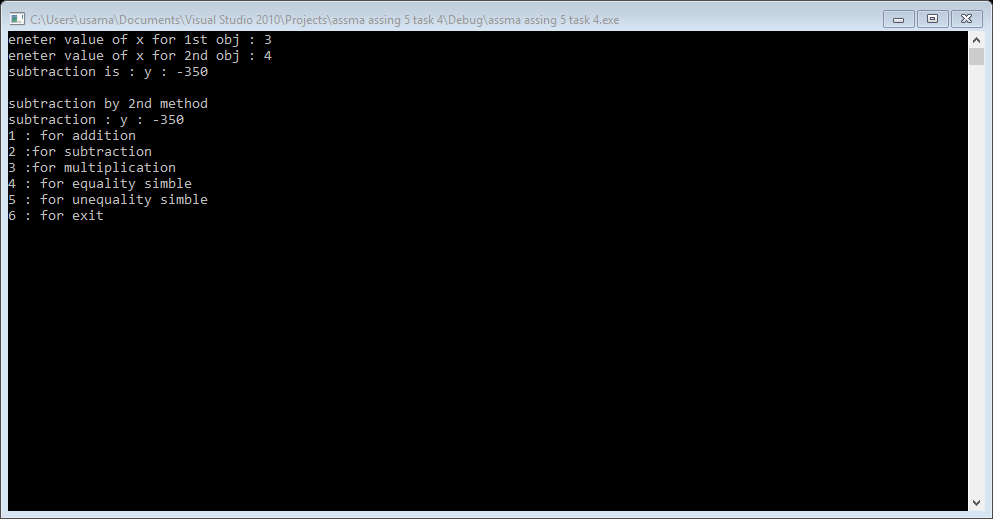
cout<<"objects are equal"<<endl;

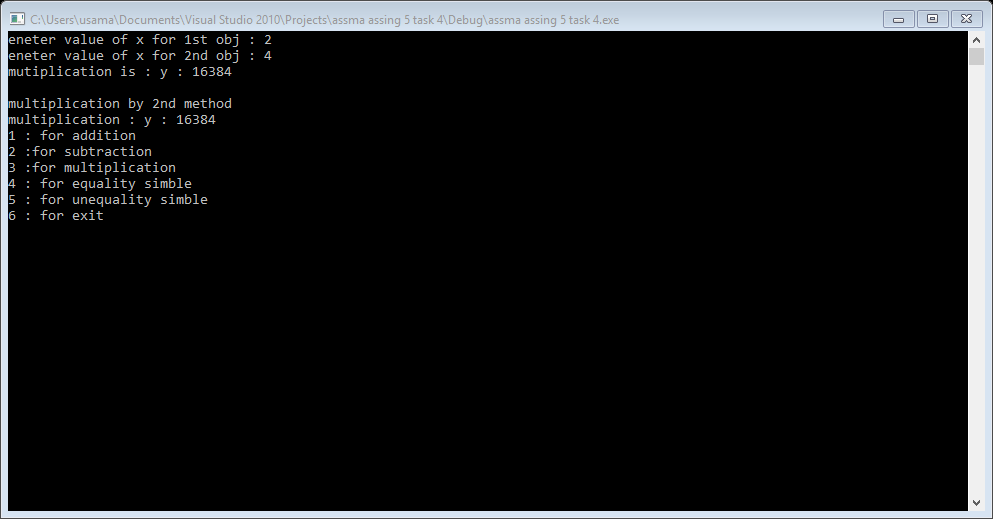
}

}

Result:







Task 5:

#include<iostream>

using namespace std;

class rational

{

private:

int n;

int d;

public:

rational()

{

}

void input()

{

cout<<"enter numerator : ";

cin>>n;

cout<<"enter denumerator :";

cin>>d;

cout<<"you enter : "<<n<<"/"<<d<<endl;

}

void show()

{

cout<<"result : "<<n<<"/"<<d<<endl;

}

rational operator +(rational r)

{

rational temp;

temp.n=r.n\*d+r.d\*n;

temp.d=d\*r.d;

temp.reduce();

return temp;

}

rational operator -(rational r)

{

rational temp;

temp.n=r.n\*d-r.d\*n;

temp.d=d\*r.d;

temp.reduce();

return temp;

}

rational operator \*(rational r)

{

rational temp;

temp.n=r.n\*n;

temp.d=r.d\*d;

temp.reduce();

return temp;

}

rational operator /(rational r)

{

rational temp;

temp.n=r.n\*d;

temp.d=r.d\*n;

temp.reduce();

return temp;

}

bool operator ==(rational r)

{

rational temp;

if(r.n==n)

{

if(r.d==d)

{

return 1;

}

else

{

return 0;

}

}

}

void operator ++()

{

n=n+1;

d=d+1;

}

void operator --()

{

n=n-1;

d=d-1;

}

void reduce()

{

/\*int x=0;

int y=0;\*/

int large=0;

int gcd=0;

cout<<"result : "<<n<<"/"<<d<<endl;

if(n>d)

{

large=n;

}

else

{

large=d;

}

for(int i=large; i>=2; --i)

{

if(n%i==0 && d%i==0)

{

gcd=i;

break;

}

}

if(gcd!=0)

{

cout<<"after reducing the fraction : ";

cout<<n/gcd<<"/"<<d/gcd<<endl;

}

}

};

int main()

{

rational r1,r2;

r1.input();

r2.input();

int c=0;

do

{

cout<<"0 : for exit "<<endl;

cout<<"1 : for addition "<<endl;

cout<<"2 : for subtraction "<<endl;

cout<<"3 : for multiplication"<<endl;

cout<<"4 : devision"<<endl;

cout<<"5 : increment "<<endl;

cout<<"6 : decrement "<<endl;

cout<<"7 : take again input"<<endl;

cin>>c;

if(c==1)

{

(r1+r2);

}

if(c==2)

{

r1-r2;

}

if(c==3)

{

r1\*r2;

}

if(c==4)

{

r1/r2;

}

if(c==5)

{

r1++;

r2++;

}

if(c==6)

{

r1--;

r2--;

}

if(c==7)

{

r1.input();

r2.input();

}

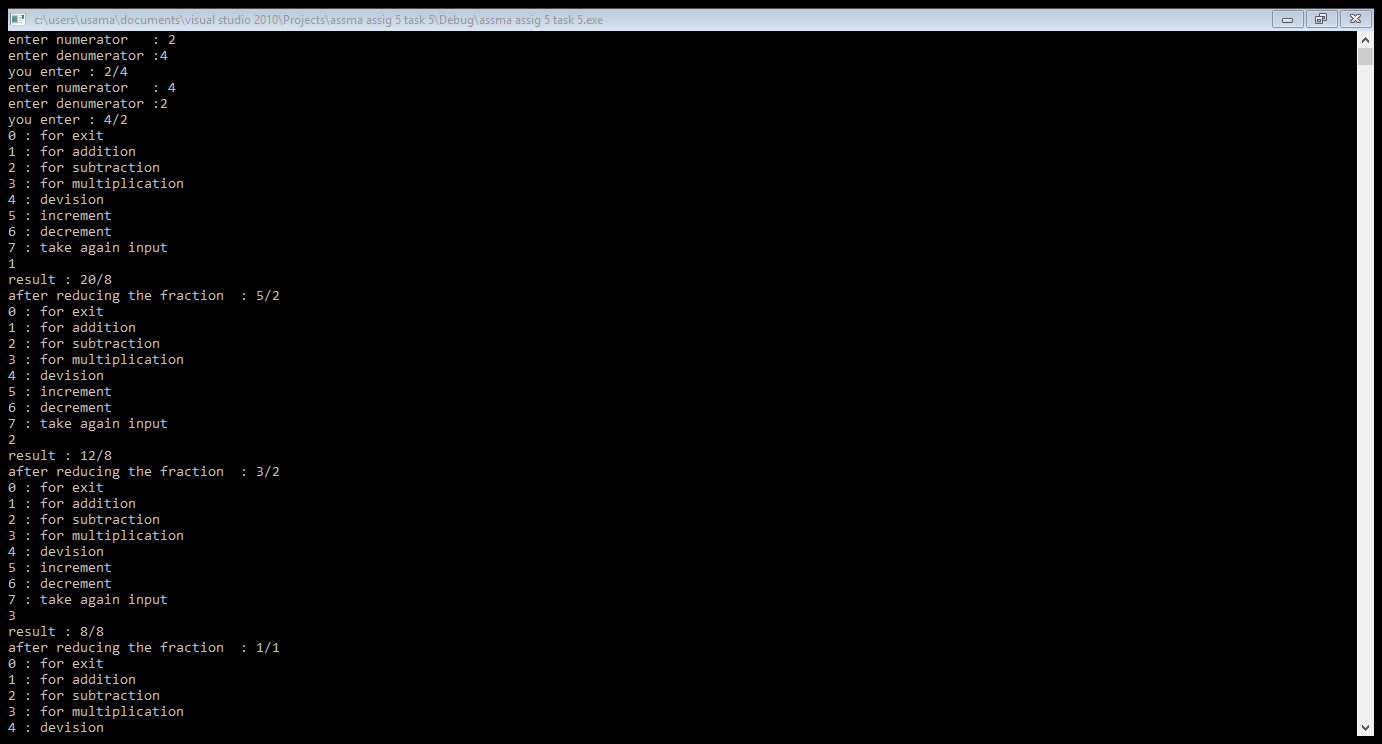
}while(c!=0);

system("pause");

return 0;

}

Result:



Task 6:

#include<iostream>

#include<string>

using namespace std;

class animal

{

public:

string name;

virtual void sound()

{

cout<<"animal sound"<<endl;

}

};

class cat:public animal

{

public:

cat()

{

name="cat";

}

void sound()

{

cout<<"name : "<<name<<" ";

cout<<"sound :"<<"meowmoew"<<endl;

}

};

class dog:public animal

{

public:

dog()

{

name="dog";

}

void sound()

{

cout<<"name : "<<name<<" ";

cout<<"sound : "<<"kakao"<<endl;

}

};

class tiger\_family :public animal

{

public:

tiger\_family()

{

name="tiger family";

}

void sound()

{

cout<<"name : "<<name<<" ";

cout<<"sound : "<<"i am in tiger\_family "<<endl;

}

};

class deer :public animal

{

public:

deer()

{

name="deer";

}

void sound()

{

cout<<"name : "<<name<<" ";

cout<<"sound : "<<"i am deer"<<endl;

}

};

class tiger:public tiger\_family

{

public:

tiger()

{

name="tiger";

}

void sound()

{

cout<<"name : "<<name<<" ";

cout<<"sound : "<<"i am tiger"<<endl;

}

};

class lion:public tiger\_family

{

public:

lion()

{

name="lion";

}

void sound()

{

cout<<"name : "<<name<<" ";

cout<<"sound : "<<"i am lion"<<endl;

}

};

class leopard:public tiger\_family

{

public:

leopard()

{

name="leopard";

}

void sound()

{

cout<<"name : "<<name<<" ";

cout<<"sound : "<<"i am leopard"<<endl;

}

};

int main()

{

animal \*a;

cat c;

dog d;

tiger\_family tf;

tiger t;

lion l;

leopard leo;

a=&c;

a->sound();

a=&d;

a->sound();

a=&tf;

a->sound();

a=&t;

a->sound();

a=&l;

a->sound();

a=&leo;

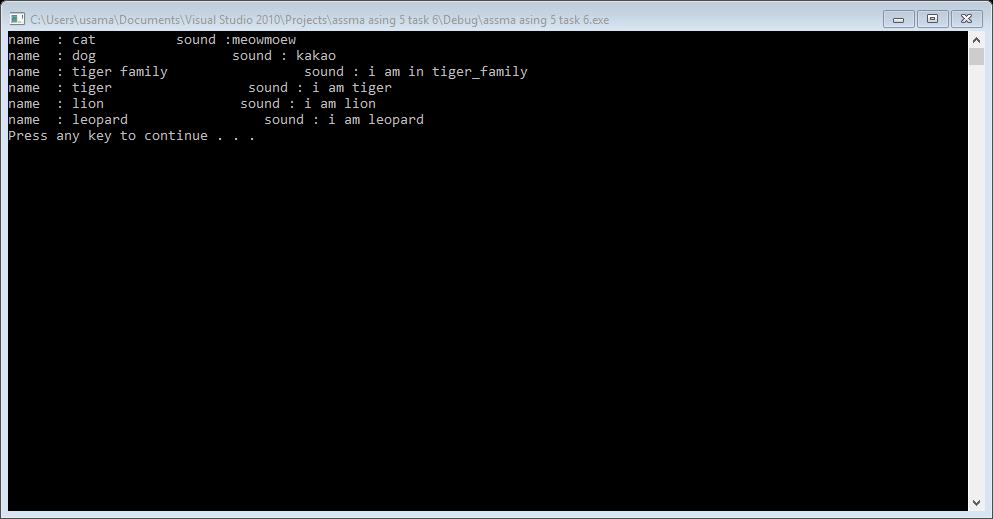
a->sound();

system("pause");

return 0;

}

Result:



Task 7:

#include<iostream>

#include<tchar.h>

using namespace std;

class polygon

{

protected:

float length;

float width;

float a;

float p;

public:

virtual float area()=0;

virtual float perimeters()=0;

virtual void display()=0;

};

class square : public polygon

{

public:

square(float l ,float w)

{

length=l;

width=w;

}

square()

{ }

float area()

{

a=length\*width;

return a;

}

float perimeters()

{

p=4\*length;

return p;

}

void display()

{

cout<<"area of square : "<<a<<endl;

cout<<"paramiters of square : "<<p<<endl;

}

};

class rectangle:public polygon

{

public:

rectangle(float l ,float w)

{

length=l;

width=w;

}

rectangle()

{ }

float area()

{

a=2\*(length\*width);

return a;

}

float perimeters()

{

p=4\*length;

return p;

}

void display()

{

cout<<"area of rectangle : "<<a<<endl;

cout<<"parameters of rectangle : "<<p<<endl;

}

};

class triangle :public polygon

{

public:

triangle(float l ,float w)

{

length=l;

width=w;

}

triangle()

{ }

float area()

{

a=1/2\*(length\*width);

return a;

}

float perimeters()

{

p=2\*length+width;

return p;

}

void display()

{

cout<<"area of triangle : "<<a<<endl;

cout<<"parameters of rectangle : "<<p<<endl;

}

};

int main()

{

int ch;

float l;

float w;

do

{

cout<<"1 :calculate area of aquare "<<endl;

cout<<"2 :caculate area of rectangle"<<endl;

cout<<"3 :caculate area of triangle"<<endl;

cin>>ch;

polygon \*ptr[3];

if(ch==1)

{

cout<<"eneter length of aquare : ";

cin>>l;

cout<<"enetr width of square : ";

cin>>w;

ptr[0]=new square(l,w);

ptr[0]->area();

ptr[0]->perimeters();

ptr[0]->display();

}

else if(ch==2)

{

cout<<"eneter length of rectangle : ";

cin>>l;

cout<<"enetr width of rectangle : ";

cin>>w;

ptr[1]=new rectangle(l,w);

ptr[1]->area();

ptr[1]->perimeters();

ptr[1]->display();

}

else if(ch==3)

{

cout<<"eneter length of triangle : ";

cin>>l;

cout<<"enetr width of triangl : ";

cin>>w;

ptr[2]=new triangle(l,w);

ptr[2]->area();

ptr[2]->perimeters();

ptr[2]->display();

}

else

{

system("cls");

ch=0;

cout<<"invalid choice"<<endl;

cout<<"please enter again ";

}

cout<<"if you calculate any other press 0 :";

cin>>ch;

system("cls");

}

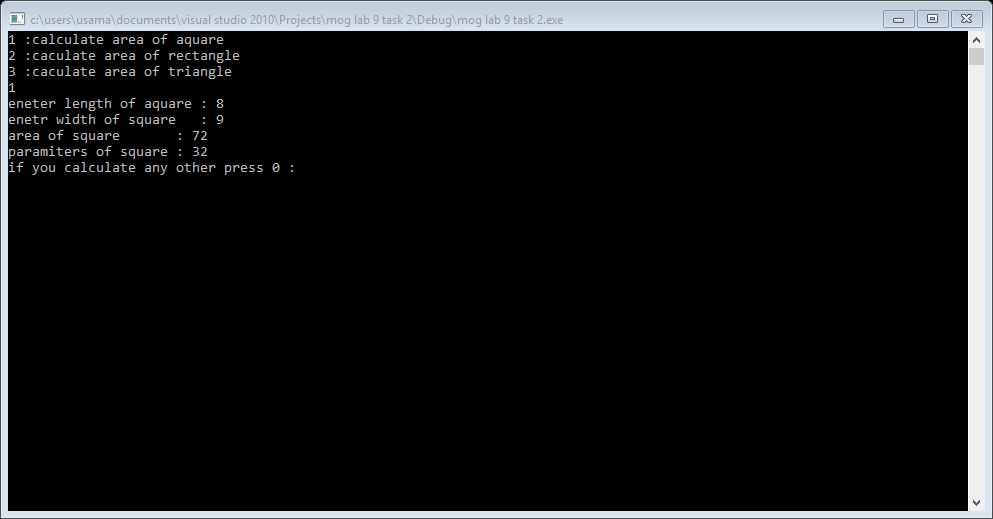
while(ch==0);

system("pause");

return 0;

}

Result:



Task 3:

#include<iostream>

#include<stdint.h>

using namespace std;

class hugeInteger

{

int32\_t no;

public:

int32\_t getno()

{

return no;

}

void input()

{

cout<<"eneter hugeno : ";

cin>>no;

}

void show()

{

cout<<"your no is : "<<no<<endl;

}

hugeInteger operator +(hugeInteger obj)

{

hugeInteger temp;

temp.no=no+obj.no;

return temp;

}

hugeInteger operator -(hugeInteger obj)

{

hugeInteger temp;

temp.no=no-obj.no;

return temp;

}

hugeInteger operator \*(hugeInteger obj)

{

hugeInteger temp;

temp.no=no\*obj.no;

return temp;

}

hugeInteger operator /(hugeInteger obj)

{

hugeInteger temp;

temp.no=no/obj.no;

return temp;

}

int operator <(hugeInteger obj)

{

if(obj.no>no)

{

return 1;

}

if(obj.no<no)

{

return 0;

}

}

int operator >(hugeInteger obj)

{

if(obj.no<no)

{

return 0;

}

if(obj.no>no)

{

return 1;

}

}

bool operator ==(hugeInteger obj)

{

if(obj.no==no)

{

return 1;

}

else

{

return 0;

}

}

bool operator !=(hugeInteger obj)

{

if(obj.no!=no)

{

return 1;

}

else

{

return 0;

}

}

};

int main()

{

hugeInteger obj1,obj2,obj3;

int c;

cout<<"0 : terminate programe "<<endl;

cout<<"1 : for addition "<<endl;

cout<<"2 : for subtraction"<<endl;

cout<<"3 : for multiplication"<<endl;

cout<<"4 : for comparison"<<endl;

cout<<"5 : for checking equality"<<endl;

cout<<"6 : again input "<<endl;

cin>>c;

obj1.input();

obj2.input();

do

{

if(c==1)

{

(obj1+obj2).show();

}

if(c==2)

{

(obj1-obj2).show();

}

if(c==3)

{

(obj1\*obj2).show();

}

if(c==4)

{

if((obj1>obj2==1))

{

cout<<"greater value : "<<obj1.getno()<<endl;

}

else

{

cout<<"greater no :"<<obj2.getno()<<endl;

}

}

if(c==5)

{

}

if(c==6)

{

}

}

while(c=!0);

system("pause");

return 0;

}

Result:

