///////////////Program1//////////////////////////

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace CSharp140716\_E1

{

class ThreeD

{

int x, y, z;

public ThreeD()

{

x = y = z = 0;

}

public ThreeD(int i, int j, int k)

{

x = i;

y = j;

z = k;

}

//Overload binary operator +

public static ThreeD operator +(ThreeD op1, ThreeD op2)

{

ThreeD result = new ThreeD();

result.x = op1.x + op2.x;

result.y = op1.y + op2.y;

result.z = op1.z + op2.z;

return result;

}

//an implicit conversion from Threed to int

public static explicit operator int(ThreeD op1)

{

return op1.x + op1.y + op1.z;

}

//Show x y z coordinates

public void show()

{

Console.WriteLine(x + ", " + y + ", " + z);

}

}

class Program

{

static void Main(string[] args)

{

ThreeD a = new ThreeD(1, 2, 3);

ThreeD b = new ThreeD(10, 10, 10);

ThreeD c = new ThreeD();

int i;

Console.Write("Here is a: ");

a.show();

Console.WriteLine();

Console.Write("Here is b: ");

b.show();

Console.WriteLine();

c = a + b;

Console.Write("Result of a + b: ");

c.show();

Console.WriteLine();

i =(int) a;

Console.Write("Result of i = (int) a: " + i);

Console.WriteLine();

i = (int)a + 2 - (int)b;

Console.Write("Result of (int)a + 2 - (int)b: " + i);

Console.WriteLine();

Console.ReadLine();

}

}

}

///////////////Program2//////////////////////////

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace CSharp140716\_E2

{

class Nybble

{

int val;

public Nybble()

{

val = 0;

}

public Nybble(int i)

{

val = i;

val = val & 0xF; //retain lower 4 bits

}

//Overload binary for Nybble + Nybble

public static Nybble operator +(Nybble op1, Nybble op2)

{

Nybble result = new Nybble();

result.val = op1.val + op2.val;

result.val = result.val& 0xF;

return result;

}

//Overload binary for Nybble + int

public static Nybble operator +(Nybble op1, int op2)

{

Nybble result = new Nybble();

result.val = op1.val + op2;

result.val = result.val & 0xF;

return result;

}

//Overload binary for int + Nybble

public static Nybble operator +(int op1, Nybble op2)

{

Nybble result = new Nybble();

result.val = op1 + op2.val;

result.val = result.val & 0xF;

return result;

}

//Overload ++

public static Nybble operator ++(Nybble op)

{

Nybble result = new Nybble();

result.val = op.val + 1;

result.val = result.val & 0xF;

return result;

}

//Overload >

public static bool operator >(Nybble op1, Nybble op2)

{

if(op1.val > op2.val)

return true;

else

return false;

}

//Overload <

public static bool operator <(Nybble op1, Nybble op2)

{

if(op1.val < op2.val)

return true;

else

return false;

}

//convert a Nybble to an int

public static implicit operator int(Nybble op)

{

return op.val;

}

//convert a an int to Nybble

public static implicit operator Nybble(int op)

{

return new Nybble(op);

}

}

class Program

{

static void Main(string[] args)

{

Nybble a = new Nybble(1);

Nybble b = new Nybble(10);

Nybble c = new Nybble();

int t;

Console.WriteLine("a: " + (int)a);

Console.WriteLine();

Console.WriteLine("b: " + (int)b);

Console.WriteLine();

//Use a Nybble in an if statement

if( a < b)

Console.WriteLine("a is less than b \n ");

Console.WriteLine();

//Add two Nybble together

c = a + b;

Console.WriteLine("c after c = a + b: "+ (int) c);

Console.WriteLine();

//Add an int to an Nybble

a += 5;

Console.WriteLine("a after a += 5: " + (int) a);

Console.WriteLine();

//Use a Nybble in an int expression

t = a \* 2 + 3;

Console.WriteLine("result of a \* 2 + 3: " + t);

Console.WriteLine();

//illustrate int assgnment and overflow

a = 19;

Console.WriteLine("result of a = 19: " + (int) a);

Console.WriteLine();

//Use a Nybble to control a lopp

Console.WriteLine("Control a for lop in Nybble");

for (a = 0; a < 4; a++)

Console.Write((int)a + " ");

Console.WriteLine();

Console.ReadLine();

}

}

}