```
/***********************************Part 7***********************************/

namespace StringThings
//aka math
{
    class StringsAKAWords
    {
        static void StringTime()
        {
            //A string variable contains a collection of characters surrounded by 
double quotes.  Essentially words/sentances/paragraphs
            //A string is an object and we can perform methods on strings like any 
other object.

            String myName = "Schaub";
            Console.WriteLine("My name is "+myName + " which has "+myName.Length+ " 
letters");

            Console.WriteLine(myName.ToUpper());
            Console.WriteLine(myName.ToLower());

            //can ad string like math +.... or using concatination command

            String myTitle = "Master";

            Console.WriteLine(myTitle + " " + myName);

            string myFullname = string.Concat(myTitle," " ,myName);

            Console.WriteLine(myFullname);

            // Can get info from string... letter type... where a certain letter is 
etc

            Console.WriteLine("The first letter in Schaub is "+ myName[0]);

            //c# starts counting at 0.... so the first letter in a name is the 0th 
letter...  this is for everything... arrays, lists, etc

            Console.WriteLine("The b in Schaub is the "+ myName.IndexOf("b")+"th 
letter");

            //some special characters like a double forward slash cause mayhem in 
strings and so need special definitions

            /*            Escape character    Result Description
                        \'           '        Single quote
                        \"           "        Double quote
                        \\           \        Backslash*/

            //Console.WriteLine("You can just throw down "Quotes"");
            Console.WriteLine("\"quotes\" need to have awkward backslashes around 
them");


            /*         Code      Result
                        \n      New Line
                        \t      Tab
                        \b      Backspace*/

            Console.WriteLine("Space\nit\nout");
            Console.WriteLine("Space\tit\tout");
            Console.WriteLine("Space\bit\bout");


        }
        static void Main(string[] args)
        {
            StringTime();
        }

    }

}

/***********************************Part 8***********************************/

namespace BooleanThings
//aka math
{
    class TrueOrFalse
    {
        static void TOrF()
        {
            //A Boolean expression returns a boolean value: True or False, by 
comparing values/variables.
            //This is useful to build logic, and find answers.

            int x = 10;
            int y = 9;
            Console.WriteLine(x > y);

            x = 10;
            Console.WriteLine(x == 10);

            Console.WriteLine(10 == 15);

            int myAge = 36;
            int votingAge = 18;
            Console.WriteLine("I am old enough to vote: "+(myAge >= votingAge));

        }
        static void Main(string[] args)
        {
            TOrF();
        }

    }

}

/***********************************Part 9***********************************/

namespace IfElseLoops
//aka math
{
    class IfElse
    {
        static void IE()
        {
/*            C# supports the usual logical conditions from mathematics:

                Less than: a < b
                Less than or equal to: a <= b
                Greater than: a > b
                Greater than or equal to: a >= b
                Equal to a == b
                Not Equal to: a != b
                You can use these conditions to perform different actions for
different decisions.

                C# has the following conditional statements:

                Use if to specify a block of code to be executed, if a specified
condition is true
                Use else to specify a block of code to be executed, if the same
condition is false
                Use else if to specify a new condition to test, if the first
condition is false
                Use switch to specify many alternative blocks of code to be
executed*/


            int myAge = 36;
            int votingAge = 18;

            if (myAge >= votingAge)
            {
                Console.WriteLine("Old enough to vote!");
            }
            else
            {
                Console.WriteLine("Not old enough to vote.");
            }

            Console.WriteLine("What is your age?");
            int strangersAge = Convert.ToInt32(Console.ReadLine());
            if (strangersAge <= 12)
            {
                Console.WriteLine("ohhhh just a weeeeee baby");
            }
            else if (strangersAge <= 19)
            {
                Console.WriteLine("ohhhhh a teenager");
            }
            else if (strangersAge <= 64)
            {
                Console.WriteLine("Just a work'n away");
            }
            else if (strangersAge <= 115)
            {
                Console.WriteLine("The golden years");
            }
            else
            {
                Console.WriteLine("LIAR!!!!!!!");
            }

        }
        static void Main(string[] args)
        {
            IE();
        }

    }

}

/***********************************Part 10***********************************/

namespace SwitchStatement
//these are great for drop down menus where the next step in your software depends
on what the user selects.
//Not the best for a console because you are dependent on what the user inputs
{
    class SwitchClass
    {
        static void Switcharooo()
        {
            Console.WriteLine("Welcome to school... what day is it today (USE:
M,Tu,W,Th,F?");
            string day = Console.ReadLine();//its angry because we are depending on
users to be good users

            switch (day)
            {
                case "M":
                    Console.WriteLine("Today's block order is ABCD.");
                    break;
                case "Tu":
                    Console.WriteLine("Today's block order is CDAB.");
                    break;
                case "W":
                    Console.WriteLine("Today's block order is BADC.");
                    break;
                case "Th":
                    Console.WriteLine("Today's block order is DCBA.");
                    break;
                case "F":
                    Console.WriteLine("Uggghhhhh Friday..... who knows...");
                    break;
                default:
                    Console.WriteLine("You clearly didn't follows the rules... No
rules! No block order!");
                    break;
            }



        }
        static void Main(string[] args)
        {
            Switcharooo();
        }

    }

}



/***********************************Part 11***********************************/

namespace WhileStatement
//these are great for drop down menus where the next step in your software depends
on what the user selects.
//Not the best for a console because you are dependant on what the user inputs
{
    class WhileClass
    {
        static void WhiledTime()
        {

            int i = 0;
            while (i < 5)
            {
                Console.WriteLine(i);
                i++;
            }
            //Do/while should be used when you want to run the code block at least
one time.
            //You should use a while loop when you don't know if you even want to
run the code

            int j = 0;
            do
            {
                Console.WriteLine(j);
                j++;
            }
            while (j < 5);

        }
        static void Main(string[] args)
        {
            WhiledTime();
        }

    }

}

/***********************************Part 12***********************************/

namespace ForStatement
//these are great for when you know exactly how many times you want something to
happen

{
    class ForClass
    {
        static void OnwardNForward()
        {

            for (int i = 1; i <= 2; ++i)
            {
                Console.WriteLine("Outer: " + i);

                // Inner loop
                for (int j = 1; j <= 3; j++)
                {
                    Console.WriteLine(" Inner: " + j);
                }
            }

            string[] counting = { "One", "Two", "Three", "Four" };
            foreach (string i in counting)
            {
                Console.WriteLine(i);
            }

        }
    static void Main(string[] args)
        {
            OnwardNForward();
        }

    }

}
```