

### Data Types

string	2 bytes/char
bool	
char	2 bytes
byte	1 byte
short	2 bytes
int	4 bytes
long	8 bytes
float	4 bytes
double	8 bytes
decimal	16 bytes

Float: Can hold up to seven significant digits of accuracy

Double: Can hold 15 or 16 significant digits of accuracy

Decimal: Has a greater precision and a smaller range. Suitable for financial and monetary calculations

### Displaying Variable Values

```
using System;
public class DisplaySomeMoney
{
    public static void
Main()
    {
        double someMoney = 39.45;

        Console.WriteLine(someMoney);
    }
}
```

### Operators

+	Addition	-	Subtraction
*	Multiplication	/	Division
%	Remainder		

### Suffixes

Put an F after a number to make it a float

float pocketchange = 4.87F;

Put a D after it to make it a double (default)

float pocketchange = 4.87D;

Put an M (money) after it to make it a decimal

float pocketchange = 4.87M;

Scientific notation

Includes an E (for exponent)

### Methods

Define it – takes 4 steps

Declaration (or method header, or signature)

```
{
Code (method body)
```

Return value;

```
}
```

Call it (invoke it)

### Method declarations

Method with no return value, and passes no parameters

```
public static void DisplayHelloText()
DisplayHelloText();
```

Method with no return value, and passes one parameter

```
public static void DisplaySalesTax(double pAmt)
DisplaySalesTax(1243.00);
```

Method with no return value, and passes two parameters

```
public static void DisplaySalesTax(double pAmt, double ptaxRate)
DisplaySalesTax(1243.00, .09);
```

Method that returns a value, and passes two parameters

### Method declarations (cont)

```
public static double DisplaySalesTax-
(double pAmt, double ptaxRate)
Amount = DisplaySalesTax(1243.00, .09);
return Answer;
```

(if no return, then void, else, return type, e.g. integer32, string, double, etc)

### Convert Methods

ToBoolean()	to an equivalent Boolean value	ToInt16()	to a signed integer
-------------	--------------------------------	-----------	---------------------

ToByte()	to an 8-bit unsigned integer	ToInt32()	to a signed integer
----------	------------------------------	-----------	---------------------

ToChar()	to a Unicode character	ToInt64()	to a signed integer
----------	------------------------	-----------	---------------------

ToDecimal()	to a decimal number	ToString()	to its equivalent String representation
-------------	---------------------	------------	---

ToDouble()	to a double-precision floating-point number	ToUInt16()	to a unsigned integer
------------	---	------------	-----------------------

### Arrays Functions

Array.Sort(arrayName)	Arranges array items in ascending order
-----------------------	---

Array.Length	Length of the array
--------------	---------------------

Array.Reverse(arrayName)	Reverses the order of items in an array
--------------------------	---



### Arrays Functions (cont)

Array.BinarySearch(ArrayName, value) Finds a requested value in a sorted array

Do not use BinarySearch() under these circumstances

If your array items are not arranged in ascending order

If your array holds duplicate values and you want to find all of them

If you want to find a range match rather than an exact match

### Array

#### Assigning values to array elements

```
double[] sales = new double [20];
sales[0] = 2100.00;
sales[1] = 3256.06;
```

#### Printing an element value

```
Console.WriteLine(sales[1]);
```

#### \*\*for loop to search a Parallel Array

```
for (int i = 0; i < nameArray.Length; i++)
{
    if (nameArray[i] == dwarfName)
    {
        do something
    }
}
```

### Multidimensional Arrays

One-dimensional or single-dimensional array Picture as a column of values. Elements can be accessed using a single subscript

Multidimensional arrays Require multiple subscripts to access the array elements

### Conditional Operators

#### Conditional AND operator

-Determines whether two expressions are both true

-Written as two ampersands (&&)

-You must include a complete Boolean expression on **each side** of the operator

Written as `If (age >= 0 && age < 120)`

#### Conditional OR operator

Used when you want some action to occur even if only one of two conditions is true

Written as `if (a == 1 || b+c > 3)`

### Loops

#### if statements

```
if (statement)
{
    do something
}
```

#### while loop

```
while (statement)
{
    do something
}
```

#### //end loop

-Definite loop or counted loop

Loop for which the number of iterations is predetermined

-Indefinite loop

Value of a loop control variable is not altered by arithmetic, but instead, is altered by user input (ask user a question inside the loop, exit based on a particular answer)

#### do-while loop

```
do
{
    do something in the loop
}
```

### Loops (cont)

`while (statement)`

#### for loop

When using a for statement, you can indicate in one place:

Starting value for the loop control variable

Test condition that controls loop entry

Expression that alters the loop control variable

`for (statement)`

`do something`

### DateTime and TimeSpan

`DateTime(yr, mo,day)` `currdt.AddDays(1)`

`DateTime(yr, mo,day, hr, min, sec)` `currdt.Subtract-Days(1)`

`DateTime.Parse("-1/1/2015")` `TimeSpan(hrs,min,sec)`

`DateTime.Now` `TimeSpan.FromSeconds(120)`