METHODS

- Methods are used in programs to eliminate repetitive code and to create an easy-to-follow, streamlined program.

- In some other languages a method is referred to as a “function” or a “subroutine”. In fact, these terms are nearly interchangeable.

- Using general methods can help divide a complex application into more manageable units of code. This helps meet goals of readability and re-usability. As you build projects, it will be obvious where such a method is needed. Look for areas in your application where code is repeated in different places. It would be best (shorter code and easier maintenance) to put this repeated code in a method. And, look for places in your application where you want to do some long, detailed task – this is another great use for a general method. It makes your code much easier to follow.

CREATING METHODS

- Here is an example method that will clear a bunch of textboxes:

```
void clearOrder()
{
    txtburger.text = "";
    txtdrink.text = "";
    txtfries.text = "";
}
```

- The 1st line tells the computer that we are creating a method called clearOrder(). The keyword void means that this method will not be returning any value to the calling code. The parentheses after the method name can include an optional list of parameters.

- After the method declaration, we have a set of curly braces. This is the body of the method and any code within the curly braces will be executed every time the program calls this method.

CALLING METHODS

- A statement that uses a method is said to be “calling” the method:

```
clearOrder()
```