CECS 174 – Lecture 18

Methods –

A method is a block of statements that are easily invoked by calling the method's name. They are a convenient way to break up your program into smaller and more manageable pieces. Every method has a definition that is made up of a header and a body. There are two types of methods, void methods, which do not return a value; and value-returning methods, which do return a value.

Method Definition –

The header is made up of modifiers, a return type, the method's name, and an input parameter list (which can be empty). The body is made up of the statements that will be performed when the method is called.

```
public static void methodX( int par1, int par2 )
{
     //statements
}
```

The code for a method definition is written inside of the class but outside of the main function.

Method Call –

The method is called using its name and any parameters it requires.

```
methodX(p1, p2);
```

A method can either be called from within the main, or from another method.

Example:

```
public class UsingMethods {
     public static void main (String [] args) {
          Scanner in = new Scanner(System.in);
          displayMenu();
          int choice = in.nextInt();
     }

     //This method displays the menu
     public static void displayMenu ( ) {
          System.out.println("Menu:");
          System.out.println("1.  Add Values");
          System.out.println("2.  Subtract Values");
     }
}
```

Void Methods – methods that do not return a value.

Example 1: method with no parameters

```
public class TestMethod {
     public static void main (String [] args) {
          int x = 10;
          System.out.println("X is " + x);
          displayY ( );
     }
    //This method declares and displays the variable y
     public static void displayY ( ) {
          int y = 3;
          System.out.println("Y is " + y);
     }
}

/* Output
X is 10
Y is 3
*/
```

The variable y is local to the method and the variable x is local to the main. The main cannot see the variable y and the displayY method cannot see the variable x.

Example 2: method with one parameter.

```
public class TestMethod {
     public static void main (String [] args) {
          int x = 10;
          System.out.println("X is " + x);
          displayNum ( x );
     }
     //This method passes in an int and displays it
     public static void displayNum ( int num ) {
          System.out.println ("Num is " + num);
     }
}

/* Output
X is 10
Num is 10
*/
```

The variable x is local to the main method and the variable num is local to the displayNum method. The variable x's value is passed to the displayNum method when it is called, the variable num is assigned that value, and then the variable num is displayed.