W06 Articulate – Encapsulation

Cecilia Michalek

Second Principle of Programing with Classes -Encapsulation

The Act of enclosing, in this case, code is called encapsulation. When creating a program, it is important to decide which behavior a specific class needs to have and then encapsulate or hide the details of how they preform those tasks or behaviors. By encapsulating specific task within a class makes it so that other parts of the program don’t see it or is unable to access it and change it.

With encapsulation you are preventing other parts of the program from accessing that data or task, but if other parts of the program need the information, it can be accessed with methods that get and sets the value to be accessed within that class.

When encapsulation is used, the class stays in control, the data is hidden, and more secure. This is one of the biggest benefits of encapsulation. It protects programmers from having their programs break.

The bottom line is that encapsulation ensures that classes are well defined, with behaviors that do what they need to do within that class. This will make the classes more purposeful and understandable.

Example of encapsulation:

**public** **class** **Pets**

{

**private** string \_name;

**private** string \_bread;

**public** string **GetPetName**()

{

**return** "Your pets name is: " + \_name;

}

**public** string **GetPetInfo**()

{

**return** \_name + " " + \_bread;

}

}

While making these member variables private we are hiding that data and controlling how other parts of the program access it. Here in this pet class, we can access the member variable info through public methods that in turn have access to that data within the class. We can use those methods to use that data in other parts of the code, also with constructor if needed.