**CSE 111 W02 Abstraction**

An abstraction is the process of turning complex ideas into simple ones. This removes the characteristics from something and only the essential ones remain. These abstractions are utilized by programming with classes. Objects, or conceptual models for a category of things- real or imagined that have specific responsibilities within the program. For example, an object holds and provides identifying information about a person. These objects have attributes or (member variables) and behaviors (methods) that allow the program to run and produce the outcome by pulling from those objects that are placed in a class. A file must be created with the class name like this, Person.cs. Now the object for this class is a Person. The identifying information needs to be tied within this class and all refers back to the object or Person. These responsibilities are things like identifying information in the form of a string variable and a behavior that places the information into the desired output. The class is like the template for the program. It is like a recipe for a desired product or food item.

Here is a class diagram:

Class Diagram

Person

\_givenName: string

\_familyName: string

ShowEasternName(): void

ShowWesternName(): void

These object state is translated into variables

called attributes. In code the string is attached.

The behaviors are translated into functions

called method.

If we are to run this within the C# code, the output would be something like this,

Person person = new Person ();

person. \_givenName – “Joseph Smith”

person. \_familyName = “Smith”

person. ShowWesternName();

output would be

Joseph Smith

If we were to put both, the ShwEasternName’s output would be, Smith, Joseph.