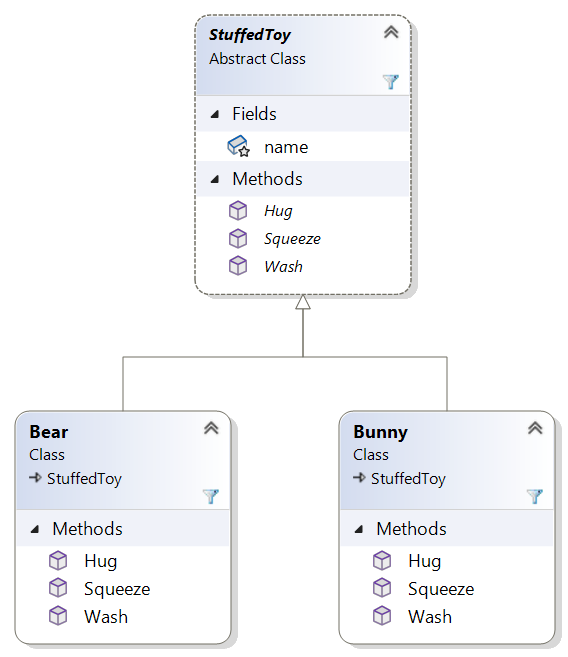
# CSCI 345 - Object Oriented Design

# Assignment 02

# Class Inheritance, Abstract Base Classes, and Runtime-Binding

# Program Specification

Mr. Pumphry, the renowned project manager of all quirky projects, has tasked you, the newly hired junior software developer, to implement the following UML Class Diagram:



1. Create an abstract class named StuffedToy that has three abstract methods named Hug that takes no parameters and has a return type of void, Squeeze that takes no parameters and has a return type of void, and Wash that takes no parameters and has a return type of void. The StuffedToy class has one protected field named name of type string. This field represents the stuffed toy’s name. Create a constructor that initializes the field.
2. Create a class named Bear that derives from StuffedToy and implements the abstract methods defined in the StuffedToy class. For each method, simply output the name of the stuffed toy, the type of stuffed toy, and whether it’s being hugged, squeezed, or washed. Create a constructor that calls the base class constructor to initialize the name field.
3. Create a class named Bunny that derives from StuffedToy and implements the abstract methods defined in the StuffedToy class. For each method, simply output the name of the stuffed toy, the type of stuffed toy, and whether it’s being hugged, squeezed, or washed. Create a constructor that calls the base class constructor to initialize the name field.
4. Create a static method named ProcessStuffedToys in the Program class. The ProcessStuffedToys method has a parameter that receives a list of StuffedToy object references. The method should call the Hug, Squeeze and Wash method for each object reference.
5. Below is a test program.

class Program

{

static void Main(string[] args)

{

List<StuffedToy> stuffedToys = new List<StuffedToy>();

StuffedToy bear = new Bear("Teddy");

StuffedToy bunny = new Bunny("Bugs");

stuffedToys.Add(bear);

stuffedToys.Add(bunny);

ProcessStuffedToys(stuffedToys);

Console.ReadLine();

}

}