# CSCI 345 - Object Oriented Design

# Assignment 03

# Interfaces 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:

Graphical user interface, application, chat or text message

Description automatically generated

1. Design interfaces named IStuffedToy. The IStuffedToy interface is defined with three methods: 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.
2. Create a class named Bear that implements the IStuffedToy interface. For each method, simply output the name of the stuffed toy and whether it’s being hugged, squeezed, or washed. The Bear class has one private field named name of type string. This field represents the stuffed toy’s name. Create a constructor that initializes the field.
3. Create a class named Bunny that implements the IStuffedToy interface. For each method, simply output the name of the stuffed toy and whether it’s being hugged, squeezed, or washed. The Bunny class has one private field named name of type string. This field represents the stuffed toy’s name. Create a constructor that initializes the field.
4. Create a static method named ProcessStuffedToys in the Program class. The method has a parameter that receives an array of objects that implement the IStuffedToy interface. The method should call each method in the IStuffedToy interface for each array element object.
5. Below is the test program.

class Program

{

static void Main(string[] args)

{

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

IStuffedToy bear = new Bear("Teddy");

IStuffedToy bunny = new Bunny("Bugs");

stuffedToys.Add(bear);

stuffedToys.Add(bunny);

ProcessStuffedToys(stuffedToys);

Console.ReadLine();

}

}