# Lab: Interfaces and Abstract Classes

## Instructions

### Part 1

1. Using C# IDE, create a program that implements an abstract class called **Animal**.

* This class has the properties: **Name**, **Colour** and **Age**.
* The class has the following methods:
  + Getter methods
  + Setters methods
  + A method called Eat that is an abstract method of type void.

1. Create a **Dog** class that implements the Animal class and the Eat method that prints “Dogs eat meat.”
2. Create a **Cat** class that implements the above Animal class and the Eat method that prints “Cats eat mice.”
3. To test the program, ask the user for a dog name and create a new Dog type object from the Main of the program. Give the Dog object a name, colour and age, and then execute the getter methods to print these properties and the Eat methods.
4. Repeat the previous step for a Cat object.

### Part 2

1. Using C# IDE, create a program that implements an **IAnimal** interface.

* The interface has the properties: Name, Colour, Height and Age.
* The interface has the following methods:
  + Getters methods
  + Setters methods
  + A method called “Eat”
  + A method called “Cry”
* The Eat method is an abstract method of type void. The Cry method is a method of type string.

1. Create a Dog class that implements the **IAnimal** interface. The Eat method should print “Dogs eat meat” and the “Cry” method should print “Woof!”
2. Create a Cat class that implements the above **IAnimal** interface. The Eat method should print “Cats eat mice” and the “Cry” method should print “Meow!”
3. To test the program, ask the user for a dog name and create a new Dog type object from the Main of the program. Then ask the user to give the Dog object a name, height, colour and age, and then execute the getter methods to print these properties and run the Eat and Cry methods.
4. Repeat the previous step for a Cat object.
5. Create a list called “animals” and add some animal objects to the list.
6. Print the names of all the animals.