# Assignment 11 - Abstraction and Polymorphism

## Title: Animal Classification System

### Objective:

Demonstrate the concepts of **Abstraction** and **Polymorphism** by creating an animal classification system using an abstract class and derived classes with polymorphic methods.

---

### Problem Statement:

Create an animal classification system where various types of animals have different sounds and ways of moving. Define the following structure:

### 1. Abstract Class: `Animal`

**Fields:**

- `name` (String) – stores the name of the animal.
- `age` (int) – stores the age of the animal.

**Constructor:**

- A constructor to initialize the `name` and `age` fields.

**Abstract Methods:**

- `makeSound()` – an abstract method to print the sound of the animal.
- `move()` – an abstract method to describe the way the animal moves.

**Non-Abstract Method:**

- `displayInfo()` – displays the animal's `name` and `age`.

---

### 2. Derived Classes:

### Class: `Dog`

- **Override** `makeSound()` – print the sound a dog makes, like "Barks".
- **Override** `move()` – print how a dog moves, like "Runs on four legs".

### Class: `Bird`

- **Override** `makeSound()` – print the sound a bird makes, like "Chirps".
- **Override** `move()` – print how a bird moves, like "Flies in the sky".

### Class: `Fish`

- **Override** `makeSound()` – print the sound a fish makes, like "Blubs" or any appropriate sound.
- **Override** `move()` – print how a fish moves, like "Swims in water".

---

## 3. Main Class

In the `main` method of a class (e.g., `Zoo`), demonstrate **Polymorphism** as follows:

- Create an array or list of `Animal` references.
- Add instances of `Dog`, `Bird`, and `Fish` to the array/list.
- Loop through the array/list and call `makeSound()`, `move()`, and `displayInfo()` on each object.

---

## Example Output:

```
Dog Info: Name: Buddy, Age: 5
Sound: Barks
Movement: Runs on four legs

Bird Info: Name: Kiwi, Age: 2
Sound: Chirps
Movement: Flies in the sky

Fish Info: Name: Goldie, Age: 1
```

```
Sound: Blubs
Movement: Swims in water
```

---

## Instructions:

1. Define the abstract class `Animal` with the specified fields and methods.
2. Implement the `Dog`, `Bird`, and `Fish` classes by inheriting from `Animal` and overriding the abstract methods.
3. In the `main` method, demonstrate polymorphic behavior by using an array or list of `Animal` references, looping through the list, and calling the respective methods.
4. Ensure the code compiles and runs without errors, and the output is formatted as in the example.

**This assignment will help you understand and implement abstraction through the `Animal` class and polymorphism when using different `Animal` types with a common interface.**