**📝 Exercise 1: Interface - Vehicle Maintenance**

**💡 Problem Statement:**

Create an interface to manage **vehicle maintenance tasks**.

**Instructions:**

1. Create an interface **IMaintenance** with methods:
   * **Service()**
   * **Repair()**
2. Implement the interface in two classes:
   * **CarMaintenance**:
     + **Service()**: Print **"Car is being serviced"**
     + **Repair()**: Print **"Car is being repaired"**
   * **BikeMaintenance**:
     + **Service()**: Print **"Bike is being serviced"**
     + **Repair()**: Print **"Bike is being repaired"**
3. In the **Main()** method:
   * Create objects of **CarMaintenance** and **BikeMaintenance** using the **IMaintenance** reference.
   * Call both methods for each object.

**📝 Exercise 2: Interface - Customer Support System**

**💡 Problem Statement:**

Create an interface to manage **customer support tickets**.

**Instructions:**

1. Create an interface **ISupportTicket** with methods:
   * **CreateTicket(string issue)**
   * **ResolveTicket(int ticketId)**
2. Implement the interface in two classes:
   * **EmailSupport**:
     + Prints **"Creating ticket via Email: [issue]"**
     + Prints **"Resolving ticket [ticketId] via Email"**
   * **PhoneSupport**:
     + Prints **"Creating ticket via Phone: [issue]"**
     + Prints **"Resolving ticket [ticketId] via Phone"**
3. In the **Main()** method:
   * Create objects of both classes using the **ISupportTicket** reference.
   * Call both methods for each object.

**📝 Exercise 3: Abstract Class - Document Management**

**💡 Problem Statement:**

Create an abstract class to manage **different document types**.

**Instructions:**

1. Create an abstract class **Document** with:
   * An **abstract method**: **PrintContent()**
   * A **concrete method**: **ShowDocumentType()** to print **"Document Type: [Type]"**
2. Create derived classes:
   * **WordDocument**: Implements **PrintContent()** to print **"Printing Word document"**
   * **PDFDocument**: Implements **PrintContent()** to print **"Printing PDF document"**
3. In the **Main()** method:
   * Create objects of both classes using the **Document** reference.
   * Call **ShowDocumentType()** and **PrintContent()** for both objects.

**📝 Exercise 4: Abstract Class - Animal Behavior**

**💡 Problem Statement:**

Create an abstract class to represent **different animal behaviors**.

**Instructions:**

1. Create an abstract class **Animal** with:
   * An **abstract method**: **MakeSound()**
   * A **concrete method**: **Eat()** to print **"Animal is eating"**
2. Create derived classes:
   * **Dog**: Implements **MakeSound()** to print **"Barks"**
   * **Cat**: Implements **MakeSound()** to print **"Meows"**
3. In the **Main()** method:
   * Create objects of both classes using the **Animal** reference.
   * Call **Eat()** and **MakeSound()** for both objects.