**Workshop Handout: Conditional Statements Using Enums in C#**

**Overview**

In this workshop, we will learn how to use conditional statements and enums in C#. Our goal is to create a simple console application to capture and display details of a pet, categorized into predefined pet types using an enum.

**Key Concepts**

1. **Enums in C#**:
   * Enums (enumerations) represent a group of constants.
   * They assign symbolic names to a set of values, improving code readability and maintainability.
2. **Conditional Statements**:
   * Conditional statements allow the program to perform different actions based on different conditions.
   * In our example, a **switch** statement is used to set the pet category based on user input.

**Program Structure**

**1. Struct Definition**

* Define a **Pet** struct to hold pet properties: ID, name, and category.

**2. Main Method Steps**

* **User Input for Pet ID and Name**:
  + Prompt the user to enter the pet ID and name.
* **Category Selection with Validation**:
  + Display a menu of pet categories.
  + Prompt the user to choose a category number.
  + Use a **do-while** loop to ensure a valid selection.
  + Use a **switch** statement to set the pet's category based on the choice.
  + If an invalid choice is made, prompt the user to re-enter a valid choice.
* **Displaying Pet Details**:
  + Once a valid category is selected, display the pet details (ID, name, category).

**3. Enum Definition**

* Define an enum **PetCategory** with values: Dog, Cat, Bird, Fish, Reptile, Other.

**Detailed Steps**

1. **Define Struct**:
   * Create a struct to store pet information including an ID, name, and category.
2. **Get User Input**:
   * Ask the user for the pet ID and name through console prompts.
3. **Category Selection**:
   * Show a list of categories.
   * Allow the user to choose a category by number.
   * Use a loop to validate the choice, repeating until a valid selection is made.
   * Implement a **switch** statement to match the choice to the correct category.
4. **Display Information**:
   * After a valid category is chosen, print out all the pet details.
5. **Enum Usage**:
   * Define an enum for pet categories to avoid using arbitrary numbers, making the code cleaner and more understandable.

**Execution Flow**

1. **Prompt for Pet Details**:
   * The program starts by asking the user to input the pet's ID and name.
2. **Category Selection**:
   * Display a menu of categories.
   * Validate the user's choice using a loop and a **switch** statement.
3. **Display Pet Information**:
   * After a valid selection, display the pet’s ID, name, and category.

**Important Notes**

* **Input Validation**:
  + Ensuring valid input is crucial for a smooth user experience.
  + The **do-while** loop ensures that users can only proceed after making a valid choice.
* **Enums for Readability**:
  + Using enums helps avoid magic numbers in the code.
  + It provides clear, meaningful names for categories, improving code clarity.
* **Structs for Organization**:
  + Using structs organizes related data into a single unit, making it easier to manage and pass around in the program.

**Conclusion**

By the end of this workshop, you should understand how to use enums and conditional statements in C# to create user-friendly console applications. Enums help make code more readable and maintainable, while conditional statements allow for flexible and dynamic program flow.