Integrated Use of Control Structures and Loops.

Objective:

By the end of this activity, you will be able to combine if-else and switch control structures with loops to solve practical coding problems. They will write simple programs that integrate these structures to handle decision-making within loops.

Step 1: Integrating If-Else with Loops

Create a program that repeatedly asks the user to input a number between 1 and 10 and ensures the number is even. The loop should continue until the user enters a valid input using an if-else statement to check the validity.

Instructions:

1. Use a do-while loop to continuously prompt the user for an even number between 1 and 10.
2. Inside the loop, use an if-else statement to validate whether the number is even and between 1 and 10.
3. If the input is valid, print the number and exit the loop using the break statement. If it's invalid, display an error message and repeat the prompt.

Code:

```
namespace UseOfControlStructuresAndLoops
{
    public class Step1IfElseWithLoops
    {
        public static void Run()
        {
            do
            {
                Console.Write("Enter an even number between 1 and 10: ");
                string input = Console.ReadLine() ?? string.Empty;

                if (int.TryParse(input, out int number))
                {
                    if (number >= 1 && number <= 10 && number % 2 == 0)
                    {
                        Console.WriteLine("Valid input: " + number);
                        break; // Exit loop if valid
                    }
                    else
                    {
                        Console.WriteLine("Invalid input. The number must be even and between 1 and 10.");
                    }
                }
                else
                {
                    Console.WriteLine("Invalid input. Please enter a numeric value.");
                }

            } while (true);
        }
    }
}
```

Step 2: Using If-Else to Evaluate Grades

Write a program that uses a for loop and an if-else structure to evaluate a list of student grades. For each grade, determine whether the student has passed or failed based on the grade value.

Instructions:

1. Define an array named grades containing a list of student grades.
2. Use a for loop to iterate over each grade in the array.
3. Inside the loop, use an if-else statement to check if each grade is greater than or equal to 65 (passing). Print "Pass" if the grade is passing and "Fail" if it is not.

Code:

```
namespace UseOfControlStructuresAndLoops
{
    public class Step2EvaluateGrades
    {
        public static void Run()
        {
            int[] grades = [85, 42, 73, 64, 90, 58, 67];
            for (int i = 0; i < grades.Length; i++)
            {
                int grade = grades[i];

                if (grade >= 65)
                {
                    Console.WriteLine("Grade: " + grade + " - Pass");
                }
                else
                {
                    Console.WriteLine("Grade: " + grade + " - Fail");
                }
            }
        }
    }
}
```

Step 3: Integrating Switch Statements with Loops

Create a program that processes multiple orders by their status. Each order can be "Pending," "Shipped," "Delivered," or "Cancelled," and the program will print a message based on the status of each order.

Instructions:

1. Define an array named orderStatuses containing different statuses: "Pending," "Shipped," "Delivered," and "Cancelled."
2. Use a loop to iterate through the list of order statuses.
3. Inside the loop, use a switch statement to print a different message based on the order's status.

Code:

```
namespace UseOfControlStructuresAndLoops
{
    public class Step3SwitchWithLoops
    {
        public static void Run()
        {
            string[] orderStatuses = ["Pending", "Shipped", "Delivered", "Cancelled", "Unknown"];

            foreach (string status in orderStatuses)
            {
                switch (status)
                {
                    case "Pending":
                        Console.WriteLine("Order is pending and will be processed soon.");
                        break;
                    case "Shipped":
                        Console.WriteLine("Order has been shipped.");
                        break;
                    case "Delivered":
                        Console.WriteLine("Order was delivered successfully.");
                        break;
                    case "Cancelled":
                        Console.WriteLine("Order has been cancelled.");
                        break;
                    default:
                        Console.WriteLine("Unknown order status: " + status);
                        break;
                }
            }
        }
    }
}
```

Step 4: Evaluating Student Grades with Switch and Loops

Write a program that uses a for loop to iterate over a list of student scores and a switch statement to assign letter grades based on the score.

Instructions:

1. Define an array of student scores.
2. Use a for loop to iterate through each score.

3. Inside the loop, use a switch statement to assign a letter grade (A, B, C, D, F) based on the score.

Code:

```
namespace UseOfControlStructuresAndLoops
{
    public class Step4GradesWithSwitch
    {
        public static void Run()
        {
            int[] scores = [95, 82, 76, 64, 58, 89, 70];

            for (int i = 0; i < scores.Length; i++)
            {
                int score = scores[i];
                string letterGrade;

                // Divide score by 10 to simplify switch logic
                switch (score / 10)
                {
                    case 10:
                    case 9:
                        letterGrade = "A";
                        break;
                    case 8:
                        letterGrade = "B";
                        break;
                    case 7:
                        letterGrade = "C";
                        break;
                    case 6:
                        letterGrade = "D";
                        break;
                    default:
                        letterGrade = "F";
                        break;
                }

                Console.WriteLine($"Score: {score} - Grade:
{letterGrade}");
            }
        }
    }
}
```

Program.cs:

```
using UseOfControlStructuresAndLoops;

class Program
{
    static void Main(string[] args)
    {
        // Call Step 1
        Step1IfElseWithLoops.Run();

        // Call Step 2
        Step2EvaluateGrades.Run();

        // Call Step 3
        Step3SwitchWithLoops.Run();

        // Call Step 4
        Step4GradesWithSwitch.Run();

    }
}
```