Developing Programs with Functions and Methods

Objective:

By the end of this activity, you will be able to develop simple programs using functions and methods in C# to solve real-world problems. You will practice defining and calling methods, passing parameters, and using return values in methods.

Step 1: Defining and Calling a Simple Method

Create a method that prints a welcome message to the console. This method demonstrates the basic structure and use of a method in C#.

Instructions:

1. Define a method called **DisplayWelcomeMessage** that prints "Welcome to the Program!" when called.
2. Call the method from the **Main** method to execute it.

Step 2: Creating a Method with Parameters

Create a method that takes a parameter to provide personalized output. This method should greet a user by name.

Instructions:

1. Define a method called **GreetUser** that takes a **string** parameter **name** and prints "Hello [name]!".
2. Call the method from the **Main** method, passing a name as an argument.

Step 3: Using Methods with Return Values

Create a method that returns a value. This method should add two numbers together and return the sum.

Instructions:

1. Define a method called **CalculateSum** that takes two integer parameters and returns their sum.

2. Store the result of the method call in a variable and print it.

Step 4: Combining Methods and Conditional Logic

Create a method that returns a boolean value based on a condition. This method should check if a number is positive.

Instructions:

1. Define a method called **IsPositive** that takes an integer parameter and returns **true** if the number is greater than zero or **false** if not.
2. Use an **if-else** statement in the **Main** method to check the result and print whether the number is positive.

Step 5: Practical Application – User Age Validation

Create a program to validate user input using methods. This program should ask for the user's age and check if they are old enough to drive.

Instructions:

1. Define a method called **IsOldEnoughToDrive** that takes an integer parameter **age** and returns **true** if the age is 18 or older.
2. In the **Main** method, prompt the user to enter their age.
3. Convert the user's input to an integer using **int.Parse()**.
4. Call the method and print whether the user is old enough to drive based on the return value of the method.

Code:

```
namespace DevelopWithFunctAndMeth
{
    public class DevelopWithFunctAndMeth
    {
        public static void Main()
        {
            // Step 1: Welcome message
            DisplayWelcomeMessage();

            // Step 2: Greet user
            string name = GetUserName();
            GreetUser(name);
```

```
            // Step 3: Sum calculation
            int num1 = ReadInteger("Enter the first number: ");
            int num2 = ReadInteger("Enter the second number: ");
            int result = CalculateSum(num1, num2);
            Console.WriteLine($"The sum of {num1} and {num2} is {result}.");

            // Step 4: Check if a user-provided number is positive
            int numberToCheck = ReadInteger("Enter a number to check if it is
positive: ");
            if (IsPositive(numberToCheck))
            {
                Console.WriteLine($"{numberToCheck} is a positive number.");
            }
            else
            {
                Console.WriteLine($"{numberToCheck} is not a positive
number.");
            }

            // Step 5: Validate user age
            int age = ReadInteger("Please enter your age: ");
            if (IsOldEnoughToDrive(age))
            {
                Console.WriteLine("You are old enough to drive.");
            }
            else
            {
                Console.WriteLine("You are not old enough to drive.");
            }
        }

        public static void DisplayWelcomeMessage()
        {
            Console.WriteLine("Welcome to the Program!");
        }

        public static string GetUserName()
        {
            Console.Write("Please enter your name: ");
            string? input = Console.ReadLine();

            while (string.IsNullOrWhiteSpace(input))
            {
                Console.WriteLine("Input cannot be empty. Try again.");
                Console.Write("Please enter your name: ");
                input = Console.ReadLine();
            }

            return input;
        }

        public static void GreetUser(string name)
        {
            Console.WriteLine($"Hello, {name}!");
        }

        public static int ReadInteger(string prompt)
        {
            Console.Write(prompt);
            string? input = Console.ReadLine();
            int number;

            while (!int.TryParse(input, out number))
            {
```

```
                Console.WriteLine("Invalid input. Please enter a valid
integer.");
                Console.Write(prompt);
                input = Console.ReadLine();
            }

            return number;
        }

        public static int CalculateSum(int x, int y) => x + y;

        public static bool IsPositive(int number)
        {
            return number > 0;
        }

        public static bool IsOldEnoughToDrive(int age)
        {
            return age >= 18;
        }
    }
}
```