Assignment 1

Submitted to:

Sir Mubashir UbaidUllah

Submitted by:

Muhammad Faizan

Registration no:

FA22-BSE-052

Course:

Visual Programming

“Department of Computer Science”

COMSATS University Islamabad,

“Vehari Campus”

Question # 1:

Event-Driven Programming vs Procedural Programming

Event-Driven Programming:

Event-driven programming is a programming paradigm where the program's execution depends on **events** like **user actions** (mouse clicks, key presses), sensor inputs, or messages from other programs. Instead of following a strict sequence, it responds whenever an event occurs.

Key Features:

- ➢ **Event Handlers:** Special functions that execute when an event happens.
- ➢ **Event Listeners:** Continuously monitor for specific events.
- ➢ **Asynchronous Execution:** The program doesn't wait for a task to finish before moving forward, making applications more responsive.

Examples:

- Clicking a **"Submit"** button triggers an event to send a form.
- Pressing **"Ctrl + S"** saves a document.
- A motion sensor detecting movement and triggering an alarm.

Procedural Programming:

Procedural programming is a step-by-step approach where code executes in a structured and sequential manner. Programs are divided into functions that follow a defined flow from start to end.

Key Features:

- ➢ **Fixed Execution Order:** The program runs line-by-line.
- ➢ **Uses Functions:** Code is divided into reusable functions.
- ➢ **Predictable Output:** Best for tasks that require strict control flow.

Conclusion:

Both paradigms have their strengths. Event-driven programming is ideal for applications requiring user interaction and real-time responsiveness, while procedural programming is better suited for structured, predictable tasks. Ethical software design means choosing the right approach based on user needs, performance, and security considerations.

Question # 2:

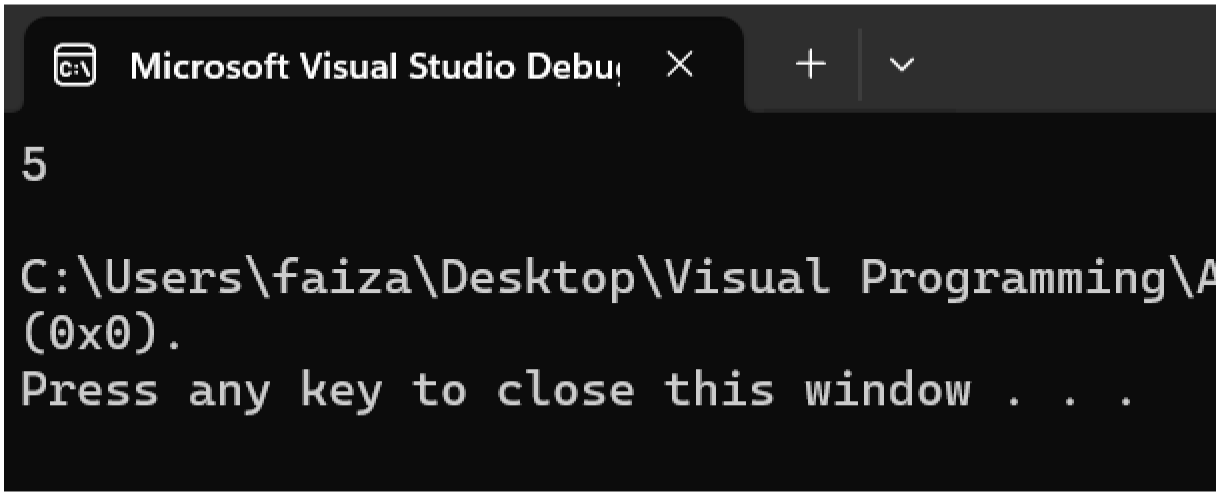
Pass by Value vs Pass by Reference

Pass by Value:

- In **pass by value**, a **copy** of the variable is passed to the method. Any changes inside the method **do not affect** the original variable.
- **Pass by value** is useful when you **do not want** a method to change the original variable.

Example:

```
using System;

class Program

{

    static void ChangeValue(int num)

    {

        num = 10;

    }
    static void Main()
    {
        int myNumber = 5;
        ChangeValue(myNumber);
        Console.WriteLine(myNumber);
    }
}
```



```
5

C:\Users\faiza\Desktop\Visual Programming\A
(0x0).
Press any key to close this window . . .
```

Explanation:

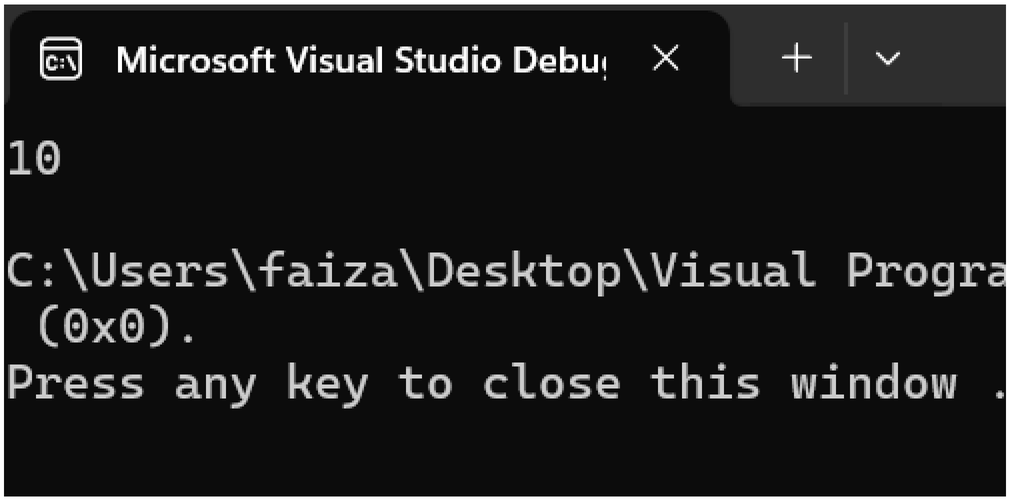
- The variable myNumber is passed as a copy to the ChangeValue method.
- Inside the method, num is modified, but this change does not affect myNumber.
- So, the output remains **5**, as the original value is not changed.

Pass by Reference:

- In **pass by reference**, the **actual variable** is passed to the method. Any change inside the method **affects the original variable** directly.
- **Pass by reference (ref)** is useful when you **want** a method to modify the original variable

Example:

```
using System;

class Program

{

    static void ChangeValue(ref int num)

    {

        num = 10;

    }

    static void Main()

    {

        int myNumber = 5;

        ChangeValue(ref myNumber);

        Console.WriteLine(myNumber);

    }

}
```



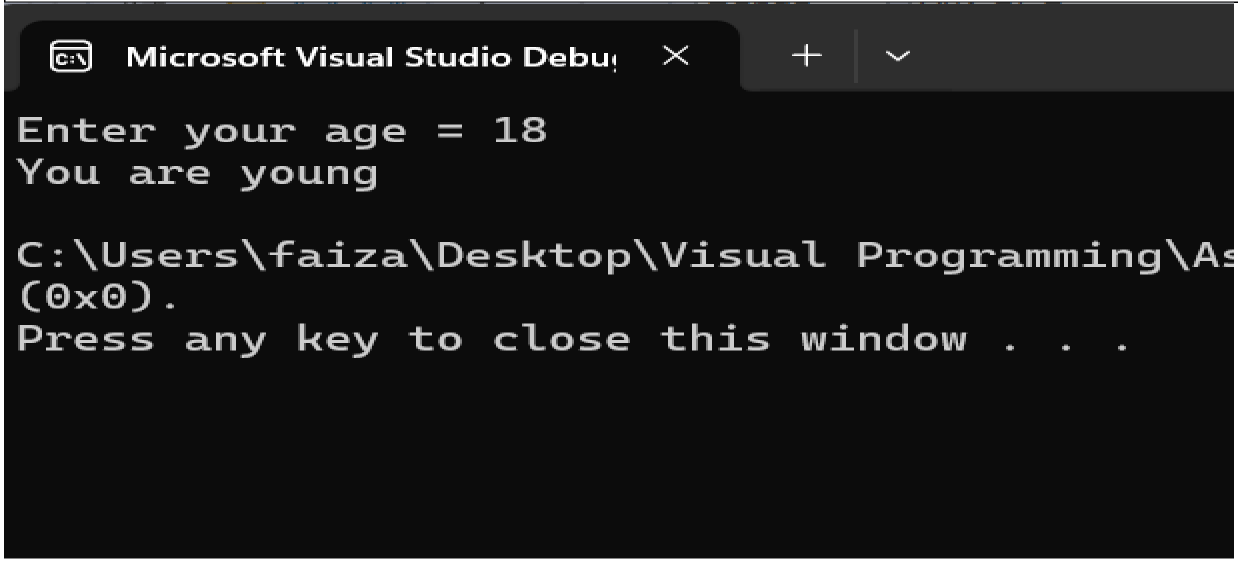
```
10

C:\Users\faiza\Desktop\Visual Progra
 (0x0).
Press any key to close this window .
```

Explanation:

- The ref keyword ensures that the **actual variable** is sent to the method.
- Inside the method, num directly refers to myNumber, so any change to num modifies myNumber.
- The output is **10**, showing that the original value has changed.

Question # 3:

```
using System;
class Program
{
    static void Main()
    {
        int age, classifier = 0;
        Console.Write("Enter your age = ");
        age = int.Parse(Console.ReadLine());
        if (age >= 0 && age <= 10)
        {
            classifier = 1;
        }
        else if (age >= 11 && age <= 17)
        {
            classifier = 2;
        }
        else if (age >= 18 && age <= 40)
        {
           classifier = 3;
        }
        else if (age >= 41)
        {
           classifier = 4;
        }
        switch (classifier)
        {
           case 1:
              Console.WriteLine("You are a child");
              break;
           case 2:
              Console.WriteLine("You are a teenager");
              break;
           case 3:
              Console.WriteLine("You are young");
              break;
           case 4:
              Console.WriteLine("You are an adult");
              break;
           default:
              Console.WriteLine("Invalid age entered.");
              break;
        }
    }
}
```



```
Enter your age = 18
You are young

C:\Users\faiza\Desktop\Visual Programming\As
(0x0).
Press any key to close this window . . .
```