

**Activity 3:** What is the cyclomatic complexity of the following piece of code?

```
public static string IntroducePerson(string name, int age)
{
    var response = $"Hi! My name is {name} and I'm {age} years old.";

    if (age >= 18)
        response += " I'm an adult.";

    if (name.Length > 7)
        response += " I have a long name.";

    return response;
}
```

**Answer:**

Cyclomatic complexity can be measured by assessing the of number of linearly independent paths within the code of the software module (IBM, 2021). The paths can be defined by the flow control statements, particularly the number of nodes and edges. It can be calculated using the below formula (GeeksforGeeks, 2024):

$M = E - N + 2$  where  $E$  = the number of edges in the control flow graph

$N$  = the number of nodes in the control flow graph

$P$  = the number of connected components

In the given code, we have four key nodes, an input response variable, conditional statement connected to age variable, a conditional statement connected to the Length attribute of name, and output response variable. We have five edges indicating the decision flow between the four variables and outcomes of the conditional statements (Figure 1). Based on the formula,

$$M = 5 - 4 + 2$$

$$M = 3$$

Cyclomatic complexity is 3.

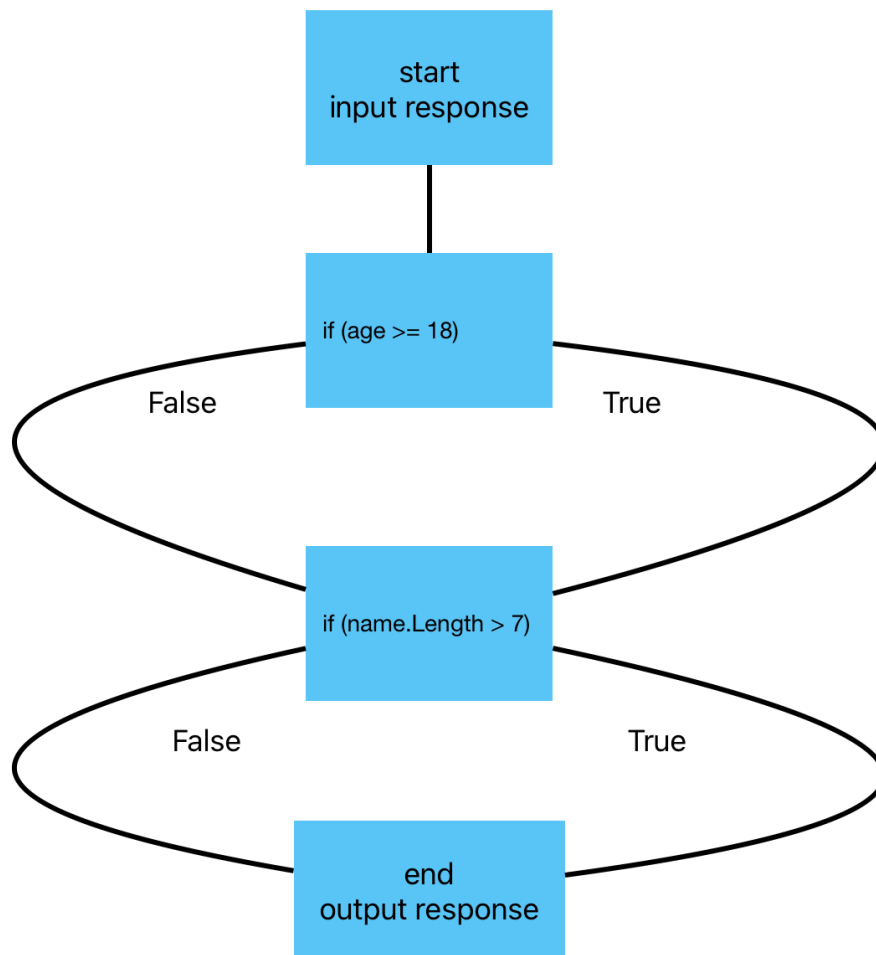


Figure 1. Nodes, edges and decision points of the activity code.

**References:**

GeeksforGeeks (2024) Cyclomatic Complexity. *GeeksforGeeks*. Available from:

<https://www.geeksforgeeks.org/cyclomatic-complexity/> [Accessed 1 June 2024]

IBM (2021) Cyclomatic complexity. *IBM Documentation*. Available from:

<https://www.ibm.com/docs/en/raa/6.1?topic=metrics-cyclomatic-complexity>

[Accessed 1 June 2024]