Programming Fundamentals Lab

Lab # 06

Nested if-else, and Switch,

Instructor: Fariba Laiq

Email: fariba.laiq@nu.edu.pk

Course Code: CL1002

Semester Fall 2023

Department of Computer Science,
National University of Computer and Emerging Sciences FAST Peshawar Campus

Nested-if in C

Nested if statements mean an if statement inside another if statement.

Syntax:

```
if (condition1)
{
  // Executes when condition1 is true
  if (condition2)
  {
    // Executes when condition1 and then condition2 is true
  }
}
```

Example 1:

```
// C program to illustrate nested-if statement
#include <stdio.h>
int main()
{
  int i = 10;
  if (i > 0) {
      // Nested - if statement
      // Will only be executed if statement above
      // is true
      if (i < 15)
          printf("i is smaller than 15\n");
      else
          printf("i is greater than 15\n");
```

```
  }
  else
     printf("i is smaller than 0\n");
  return 0;
}
```

Output

i is smaller than 15

Now let’s create a menu-driven program that uses nested if-else

Below is a C code defines a simple menu-driven program that allows the user to select from three options using nested if-else statements.
Each option corresponds to a different functionality:
checking if a triangle is equilateral, determining if a number is even or odd, and checking if the user is eligible for voting based on their age.
Here's a breakdown of how the program works:

- The user is presented with a menu displaying three options: checking if a triangle is equilateral, determining if a number is even or odd, and checking voting eligibility based on age.
- The user is prompted to enter their choice by inputting an integer, which is stored in the choice variable using the scanf function.
- The program uses an if-else if-else structure to determine which option the user has selected based on the value of choice.
- If the user selects option 1 (equilateral triangle check), the program proceeds to ask for the lengths of the three sides of the triangle (sideA, sideB, and sideC) and checks whether they are equal. If they are equal, it prints "It's an equilateral triangle"; otherwise, it prints "Not an equilateral triangle."
- If the user selects option 2 (even or odd check), the program prompts the user to enter a number (stored in the no variable) and checks if it's even or odd using the modulo operator (%). If the number is even, it prints "Even no"; if it's odd, it prints "Odd no."
- If the user selects option 3 (voting eligibility check), the program asks for the user's age (stored in the age variable) and checks if the age is greater than or equal to 18. If the age is 18 or older, it prints "You are eligible for voting"; otherwise, it prints "You are not eligible for voting."
- If the user selects any option other than 1, 2, or 3, the program prints "Invalid choice."

The if-else and if-else if-else statements are used to handle different cases or choices, making the program flow logically based on the user's input. This is an example of a basic menu-driven program implemented using nested if-else statements to handle multiple options and scenarios.

```
#include<stdio.h>
int main()
{
	printf("Menu: ");
	printf("\n1. Check if the triangle is an equilatral triangle ");
	printf("\n2. Check if a no is even or odd ");
	printf("\n3. Check if you are eligibel for voting\n");
	int choice;
	scanf("%d", &choice);
	if(choice==1)
	{
		int sideA, sideB, sideC;
		printf("Enter the side A of triangle: ");
		scanf("%d", &sideA);
		printf("Enter the side B of triangle: ");
		scanf("%d", &sideB);
		printf("Enter the side C of triangle: ");
		scanf("%d", &sideC);
		if(sideA==sideB && sideB==sideC)
		{
			printf("\nIts an equilatoral triangle");
		}
		else
		{
			printf("\nNot an equilatoral triangle");
		}
	}
```

```
        else if(choice==2)
        {
                int no;
                printf("\nEnter a no: ");
                scanf("%d", &no);
                if(no%2==0)
                {
                        printf("\nEven no");
                }
                else
                {
                        printf("\nOdd no");
                }
        }
        else if(choice==3)
        {
                int age;
                printf("\nEnter your age: ");
                scanf("%d", &age);
                if(age>=18)
                {
                        printf("\nYou are eligible for voting");
                }
                else
                {
                        printf("\nYou are not eligible for voting");
                }
        }
        else
        {
                printf("\nInvalid choice");
        }
}
```

C switch Statement

The switch statement allows us to execute one code block among many alternatives.

You can do the same thing with the if...else..if ladder. However, the syntax of the switch statement is much easier to read and write.

Syntax of switch...case

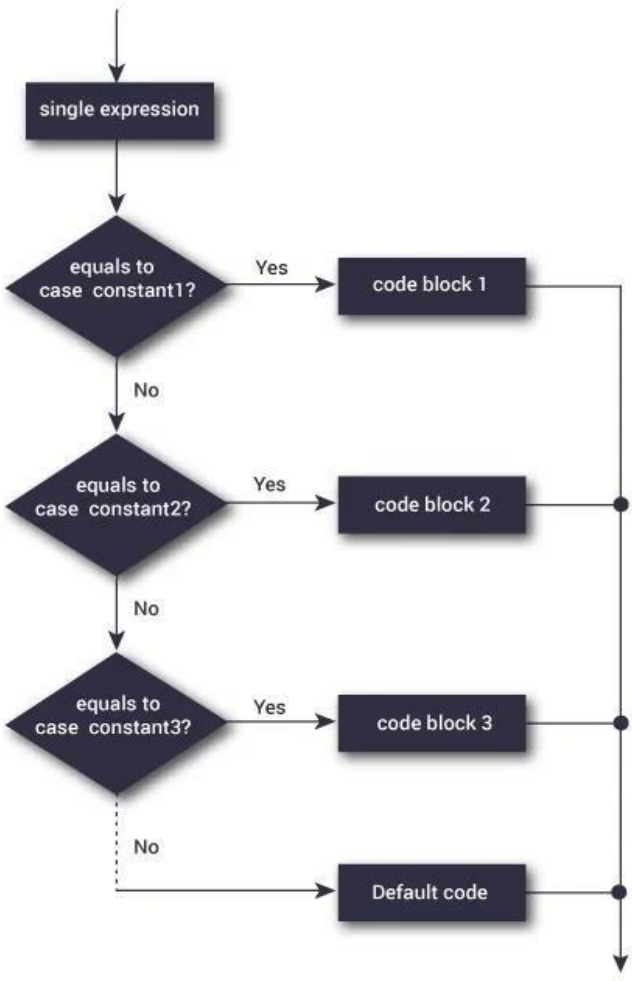
```
switch (expression)
{
  case constant1:
    // statements
    break;
  case constant2:
    // statements
    break;
  .
  .
  default:
    // default statements
}
```

How does the switch statement work?

The expression is evaluated once and compared with the values of each case label.

- If there is a match, the corresponding statements after the matching label are executed. For example, if the value of the expression is equal to constant2, statements after case constant2: are executed until break is encountered.
- If there is no match, the default statements are executed.

Switch Statement Flowchart



Example 2: Simple Calculator

```
// Program to create a simple calculator
```

```
#include <stdio.h>
int main() {
   char operation;
   int n1, n2;
   printf("Enter an operator (+, -, *, /): ");
   scanf("%c", &operation);
   printf("Enter two operands: ");
   scanf("%d %d",&n1, &n2);
   switch(operation)
   {
      case '+':
         printf("%d + %d = %.d",n1, n2, n1+n2);
         break;
      case '-':
         printf("%d - %d = %d",n1, n2, n1-n2);
         break;
      case '*':
         printf("%d * %d = %d",n1, n2, n1*n2);
         break;
      case '/':
         printf("%d / %d = %d",n1, n2, n1/n2);
```

```
            break;
        // operator doesn't match any case constant +, -, *, /
        default:
            printf("Error! operator is not correct");
    }
    return 0;
}
```

Output

```
Enter an operator (+, -, *, /): *
Enter two operands: 4 3
4 * 3 = 12
```