# Unit 5
# Control Statement

## Control Statements

- Decision-making statements in programming languages decide the direction of the flow of program execution.

## Types of Control Statements in C

- Decision-making statements available in C are:
  1. Decision-making control statements.
  2. Conditional statements
  3. Goto statements in C
  4. Loop control statements in C

## Decision-Making Control Statements

- The decision making control statement in C is used to perform the operations based on some specific condition.
- The operations specified in if block are executed if and only if the given condition is true.
- There are the following variants of if statement in C language.
  - Simple if statement
  - If-else statements
  - Nested if-else statements
  - else-if ladder

### Simple if statement

*Syntax:*

```
if(condition)
{
        Statement/s;
}
```

***Example:*** Program that identifies whether the given number is even or odd.
***Example:*** Program find the greatest between two numbers.

### if-else statements

*Syntax:*

```
if(condition)
{
        Statement/s;
}
else
```

```
{
        Statement/s;
}
```

## Nested if else Statement

*Syntax*

```
if (Condition1)
{
        if(Condition2)
        {
                Statement1;
        }
        else
        {
                Statement2;
        }
}
else
{
        if(Condition3)
        {
                Statement3;
        }
        else
        {
                Statement4;
        }
}
```

***Example:*** Program find the greatest among three numbers.

## else-if ladder

*Syntax*

```
if (condition1)
{
        Statement1;
}
else if(condition2)
{
        Statement2;
}
else if (condition3)
{
```

```
        Statement3;
    }
    .
    .
    else
    {
        //These statements would execute if all the conditions return false.
    }
```

***Example:*** Program find the greatest among three numbers.

## Conditional Control (switch) Statement

- The switch statement in C is an alternate to if-else-if ladder statement which allows us to execute multiple operations for the different possible values of a single variable called switch variable.

*Syntax*

```
    switch(expression)
    {
        case constant-expression :
            statement(s);
            break;          /* optional */

        case constant-expression  :
            statement(s);
            break;          /* optional */
        .
        .
        .

        default :           /* Optional */
            statement(s);
    }
```

***Example:*** Program that find the grade of marks any student.

## Decision Making and Looping

- A loop statement allows us to execute a statement or group of statements multiple times.
- There are mainly two types of loops in C Programming:
    - *Entry Controlled loops:* In Entry controlled loops the test condition is checked before entering the main body of the loop. For Loop and While Loop is Entry-controlled loops.
    - *Exit Controlled loops:* In Exit controlled loops the test condition is evaluated at the end of the loop body. The loop body will execute at least once, irrespective of whether the condition is true or false. do-while Loop is Exit Controlled loop.

*While Loop*

The while loop loops through a block of code as long as a specified condition is true:

*Syntax*

```
while (condition)
{
        // code block to be executed
}
```

***Example 1:*** Program to find the sum of numbers from 1 to 10.

***Example 2:*** Program to find even numbers from 1 to 100.

## For Loop

- A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

*Syntax*

```
for ( init; condition; increment )
{
  statement/s;
}
```

***Example1:*** Program to find the sum of numbers from 1 to 10.

***Example 2:*** Program to find even numbers from 1 to 100.

## do-while loop

- The do...while loop in C programming checks its condition at the bottom of the loop.
- A do...while loop is similar to a while loop, except the fact that it is guaranteed to execute at least one time.

*Syntax*

```
do {
        statement/s;
} while( condition );
```

***Example 1:*** Program to find the sum of numbers from 1 to 10.

***Example 2:*** Write a menu program using do while and switch statement.

## Goto Statements in C

- A goto statement in C programming provides an unconditional jump from the 'goto' to a labeled statement in the same function.

*Syntax*

```
goto label;
.
.
.
label: statement;
```

***Example:*** Program to demonstrate the concept of goto statement in C.

```
#include<stdio.h>
int main()
{
        int a = 0;
        label: a++;
        printf("\nValue of a is %d", a);
        if(a <= 10)
                goto label;
        return 0;
}
```

***Output:***

```
Value of a is 1
Value of a is 2
Value of a is 3
Value of a is 4
Value of a is 5
Value of a is 6
Value of a is 7
Value of a is 8
Value of a is 9
Value of a is 10
Value of a is 11
```

## Exit function

- The exit() function is used to terminate a process or function calling immediately in the program.

*Syntax*

```
exit ( int status);
```

***Example:*** Program to demonstrate the concept of exit function.

```
#include <stdio.h>
#include <stdlib.h>
int main ()
{
        printf("Start of the program....\n");
        printf("Exiting the program....\n");
        exit(0);
        printf("End of the program....\n");
        return(0);
}
```

***Output:***

```
Start of the program....
Exiting the program....
```

## Break and Continue

- The break statement can also be used to jump out of a loop.
- The continue statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

***Example:*** Program to demonstrate the concept of break and continue statements.

## Exercise

1. Discuss different types of if statements with example of each. Differentiate if statement with switch statement. (10) [TU 2074]
2. Write a program that computes the sum of digits of a given integer number. (5) [TU 2074]
3. What is looping statement? Discuss different looping statements with suitable example of each. (10) [TU 2075]
4. What is break statement? Discuss with example. How the break statement is different from continue statement? (5) [TU 2075]
5. Write a program to check whether a number entered is even or odd.(5) [TU 2075]
6. Write a program to calculate sum of first 10 odd numbers. (5) [TU 2075]
7. What do you mean by looping? Explain while loop with suitable example. Compare while loop with do-while loop. Write a program to find sum and average of first n natural numbers. (10) [TU 2077]
8. Write a program to display first n prime numbers. (5) [TU 2077]
9. What do you mean by jump statement? Explain each jump statement with example. Write a program to check whether a number entered is prime or not. (10) [TU 2078]
10. Write a program to print largest among three numbers entered by the user. (5) [TU 2078]

11. What is the difference between exit(0) and exit(1)? Discuss the need of nested structure with an example. Write a program to find the value of xy without using POW code. (10) [TU 2079]
12. Write a program to demonstrate the following menu-driven program. The user will provide an integer and alphabet for making choice and the corresponding task has to be performed according as follow:
    A. Find Odd or Even
    B. Find Positive or Negative
    C. Find the Factorial value
    D. Exit

    The choice will be displayed until the user will give “D” as a choice. (10) [TU 2079]