

Unit III

1. Give the differences between sequence control, decision control and looping control statements. Give example.

Ans: control statements help to jump control from one part of the program to another.

- Sequence control: Normally instructions in the program are executed in the same order in which they appear in the program
- Decision control: In some cases, we need to execute a set of instructions based on certain criteria. This way of controlled execution of statements can be achieved by using decision statements in the program.
- Looping control statements: they are used to repeat the execution of a sequence of statements (statement block) as long specified condition remains true.

2. Elaborate the working of If-Else statement in C with syntax and Example

Ans:

If-else statement
syntax →

```
if ( expression )  
    program statement 1  
else  
    program statement 2
```

example:

```
// Program to categorize a single character that is entered at  
the terminal  
  
int main (void)  
{  
    char c;  
    printf ("Enter a single character:\n");  
    scanf ("%c", &c);  
    if ( (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') )  
        printf ("It's an alphabetic character.\n");  
    else if ( c >= '0' && c <= '9' )  
        printf ("It's a digit.\n");  
    else  
        printf ("It's a special character.\n");  
    return 0;  
}
```

3. Explain the if-else ladder with an example

Ans:

syntax:

```
if ( expression 1 )  
    program statement 1  
else  
    if ( expression 2 )  
        program statement 2  
    else  
        program statement 3
```

both are same

```
if ( expression 1 )  
    program statement 1  
else if ( expression 2 )  
    program statement 2  
else  
    program statement 3
```

⇒ improves readability

4. What is a ternary operator? Give example

Ans:

5. Give the differences between if-else and switch case

Ans:

syntax →	alternate
<pre>switch (expression) { case value1: program statement program statement ... break; case value2: program statement program statement ... break; ... case <u>valuen</u>: program statement program statement ... break; default: program statement program statement ... break; }</pre>	<pre>if (expression == value1) { program statement program statement ... } else if (expression == value2) { program statement program statement ... } ... else if (expression == <u>valuen</u>) { program statement program statement ... } else { program statement program statement ... }</pre>

6. With relevant example give the differences between do-while and while loop

Ans:

While Statement :

Syntax →

```
while ( expression )  
    program statement (or statements)
```

ex.

```
int main(void)  
{  
    int count = 1;  
    while (count<=5)  
    {  
        printf("%i\n",count);  
        ++count;  
    }  
  
    return 0;  
}
```

② //program to reverse an entered number

```
int main(void)  
{  
    int num,rev_num;  
    printf("enter the number to be reversed:");  
    scanf("%i",&num);  
    while (num != 0)  
    {  
        rev_num= num %10;  
        printf("%i",rev_num);  
        num/=10;  
    }  
    printf("%i is the reversed number",rev_num);  
    return 0;  
}
```

→ gives blank
output if
0 is
entered

Do statement

```
do  
    program statement (or statements)  
while ( loop_expression );
```

runs minimum one time.

```
int main ()  
{  
    int number, right_digit;  
    printf ("Enter your number.\n");  
    scanf ("%i", &number);  
    do {  
        right_digit = number % 10;  
        printf ("%i", right_digit);  
        number = number / 10;  
    }  
    while ( number != 0 );  
    printf ("\n");  
    return 0;  
}
```

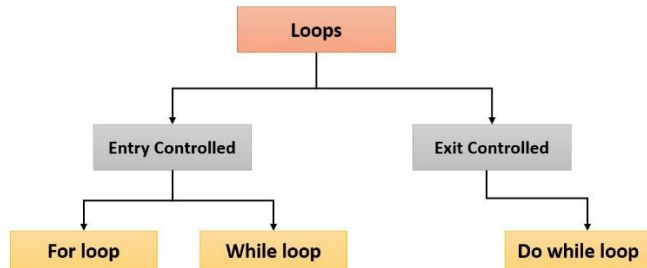
using 'do',
output will
not be blank
instead 0
is shown.

7. Discuss the role of goto, continue and break and detail

Ans:

- break is used to terminate the loop prematurely in which it appears.
- continue is used to skip the iteration in the loop based on the condition.
- goto is used for unconditional transfer of control from goto statement to a labelled statement in the program.

8. What are the different looping structures in C? Give examples



Ans:

9. Summarize the differences between Entry controlled and Exit controlled loop

Ans:

- Key difference between entry controlled Loop and exit controlled Loop is that
- Entry loop, loop block won't execute if test expression returns false.
- Exit loop, loop block will be executed even though test expression returns false.

Definite loop: when no of iterations is known. Eg: for loop

Indefinite loop: when no of iterations is not known. Eg: while, do while.

10. Explain the working of nested for loop in detail

Ans:

```
int main(void)
{
    int n, num, trinum, counter;
    for (counter = 1; counter <= 5; ++counter)
    {
        printf("what triangular number do you want?");
        scanf("%i", &num);
        trinum = 0;
        for (n = 1; n <= num; ++n)
            trinum += n;

        printf("triangular number %i is %i\n", num, trinum);
    }
    return 0;
}
```

11. Define an array? How do you declare and initialize a one-dimensional array in C?

Ans:

- It is a collection of similar data elements, and all have the same data type.
- The elements of the array are stored in consecutive memory locations and are referenced by an index (subscript).
- Declaring an array: `int marks[10];`

- Initializing array elements: `int marks[5] = { 25, 99, 76, 54, 88 }`
- Access the elements of an array:
 - `int i, marks[10];`
 - `for (i=0;i<10;i++)`
`marks[i] = -1;`
-

12.Explain the working of two-dimensional arrays with an example

Relevant programming examples to be practised