**Introduction of C language.**

↳ C is a general-purpose, procedural, and middle-level programming language used for developing computer software, system programming, applications, games, and more.

↳ It was created for programming the UNIX operating system.

↳ Example:

```
#include <stdio.h>

int main()
{
   printf ("Hello World!");
   return 0;
}
```

- **Operator and Operand from copy**
- **String from copy**
- **if Statement**

↳ The if in C is the simplest decision-making statement. It consists of the test condition and a block of code that is executed if and only if the given condition is true. Otherwise, it is skipped from execution.

↳ Syntax:

```
if (condition) {
  // block of code to be executed if the condition is true
}
```

↳ Example:

```
#include <stdio.h>
void main() {
  if (20 > 18) {
    printf("20 is greater than 18");
  }
}
```

- **else statement**

↳ The else statement is used to specify a block of code to be executed if the condition is false.

↳ Syntax

```
if (condition)
   {
    // Code to execute if condition is true
   }
   else {
   // Code to execute if condition is false
   }
```

↳ Example:

```
#include <stdio.h>
void main() {
   int n = 10;

   if (n > 5) {
      printf("%d is greater than 5",n);
   }
     else {
      printf("%d is less than 5",n);
   }

   return 0;
}
```

- **else if statement**
  - ↳ The else if statement is used to specify a new condition if the first condition is false.
  - ↳ Syntax:

```
if (condition1) {
  // block of code to be executed if condition1 is true
}
else if (condition2) {
  // block of code to be executed if the condition1 is false and
condition2 is true
}
else {
  // block of code to be executed if the condition1 is false and
condition2 is false
}
```

  - ↳ Example:

```
#include <stdio.h>
void main() {
    int n = 10;
    if (n > 0) {
        printf("Positive.\n");
    } else if (n < 0) {
        printf("Negative.\n");
    } else {
        printf("Zero.\n");
    }
}
```

- **FOR LOOP**

In C programming, the (for loop) is used to repeatedly execute a block of code as many times as instructed.
It uses a variable (loop variable) whose value is used to decide the number of repetitions.
It is generally used when we know how many times we want to repeat the code.

```
#include <stdio.h>
int main() {
  int i;
  for (i = 0; i < 5; i++) {
    printf("%d\t", i);
  }
  return 0;
}
```

- **WHILE LOOP**
  - ↳ The while loop in C allows a block of code to be executed repeatedly as long as a given condition remains true. It is often used when we want to repeat a block of code till some condition is satisfied.
  - ↳ Syntax :

```
While (condition)
{
// body
  updation
 }
```

↳ Example:

```
#include <stdio.h>
void main() {
  int i = 0;

  while (i < 5) {
    printf("%d\n", i);
    i++;
  }
}
```

- **Array**

↳ An array in C is a fixed-size collection of similar data items stored in contiguous memory locations.

↳ It can be used to store the collection of primitive data types such as int, char, float, etc., and also derived and user-defined data types such as pointers, structures, etc.

↳ It is a simple and fast way of storing multiple values under a single name.

```
#include <stdio.h>
int main()
{
   int [5];
   char [5];
   return 0;
}
```

- **Break**

↳ It was used to "jump out" of a switch statement.

↳ The break statement can also be used to jump out of a loop.

↳ This example jumps out of the for loop when i is equal to 4:

↳ Example:

```
 #include <stdio.h>
void main() {
  int i;
  for (i = 0; i < 10; i++) {
    if (i == 4) {
      break;
    }
    printf("%d\n", i);
  }
}
```