E-MODULE

C Programming Language

SUBMITTED BY:
RANI CHANDI
(DEPARTMENT OF COMPUTER SCIENCE AND IT)
CLASS:-BSC(IT)- I SEM
SUBJECT:-INTRODUCTION TO PROGRAMMING-C

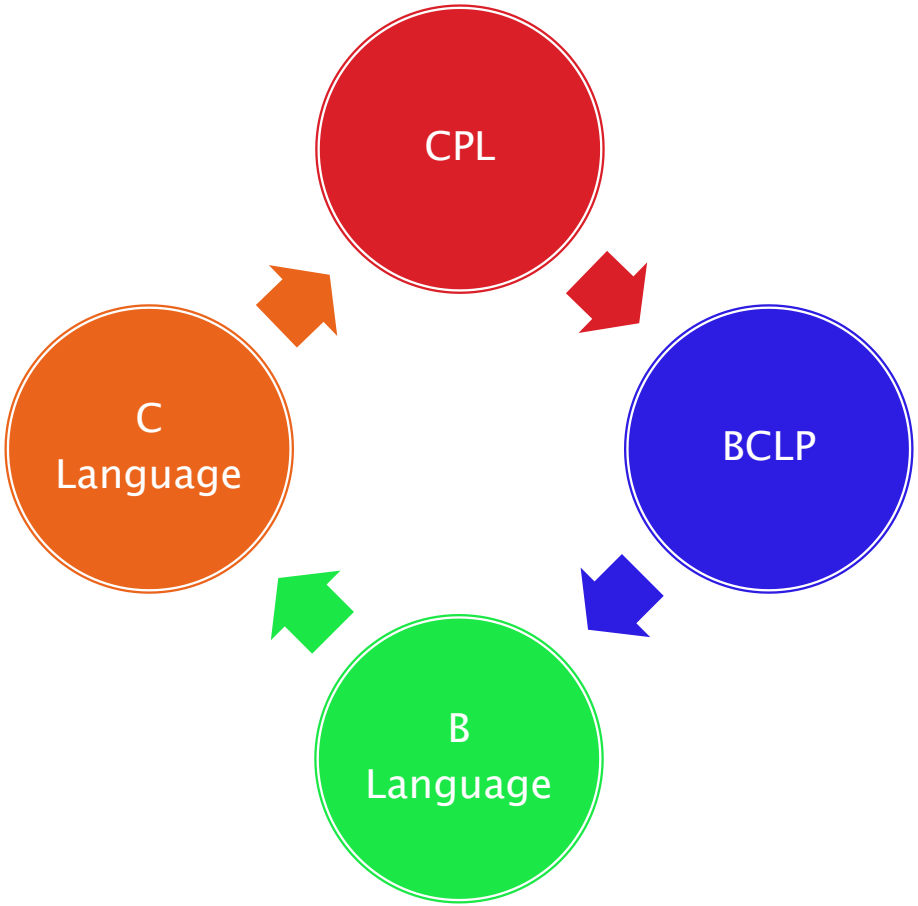
Introduction

- C is a general-purpose, procedural, computer programming language developed in 1972 by Dennis Ritchie at the Bell Laboratories to develop the UNIX operating system.

- C is the most widely used computer language.

- It is also popular and most widely used among modern software programmers.

EVOLUTION



Features

1) Easy to use Language.
2) Structured Programming Language .
3) It produces efficient programs.
4) It can handle low-level activities.
5) It can be compiled on a variety of computer platforms.
6) Portable.

Basic Structure of a ‘C’ Program

-Documentation Section

-Preprocessor Directives
-Symbolic Constants

-Global Declaration

```
main()
{
Local Declaration
Executable Statements
}

Subprogram Section
{
Local Declaration
Executable Statements
}
```

Example of C Program

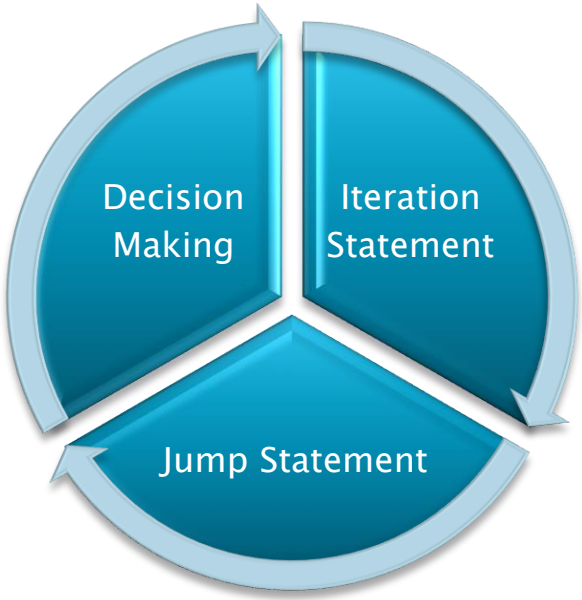
```
#Program to print a statement
main()
{
printf("Hello Students");
getch();
}
```

Control Statements

Control statements provide us the ability to specify the flow of program control, i.e. the order in which the instructions in a program will be executed. They make it possible to make decisions, to perform tasks repeatedly or to jump from one section of code to another.

There are three types of control statements in C:

-Decision making statements
-Iteration statements
-Jump statements



Decision Making

Decision-making structure require programmers to specify one or more conditions to be tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and other statements to be executed if the condition is determined to be false.

if Statement:-An if statement consists of a boolean expression followed by one or more statements.

Syntax:-The syntax of an 'if' statement in C programming language is:

```
if(condition)
{
 statement(s);
}
```

Example

```
void main()
{
int a=10,b=10;
if(a==b)
{
printf("Both are same");
}
printf("\n End of the Program");
getch();
}
```

OUTPUT:-
Both are same
End of the Program

if-else:-An if statement can be followed by an else statement, which is executed when the Boolean expression is false.

Syntax:-The syntax of an 'if-else' statement in C programming language is:

```
if(condition)
{
 statement(s);
}
else
{
statement(s);
}
```

Example

```
void main()
{
int a=10;
if(a>=7)
{
printf("Result is pass in test");
}
else
{
printf("\n not pass");
}
getch();
}
```

OUTPUT:-
Result is pass in test

Nested if:- By using nested if one can apply multiple conditions on one single situation.

Syntax:-The syntax of an 'Nested if' statement in C programming language is:

```
if(condition)
{  if(condition)
   statement(s);
   else
   statement(s);
 }
else
{
statement(s);
}
```

Example

```
void main()
{
int a=10,b=5;
if(a==10)
{ if(b==10)
 printf("both have same value");
 else
 printf("b is less than 10");
}
else
{
printf("both having different value");
}
getch();
}
```

OUTPUT:-

b is less than 10

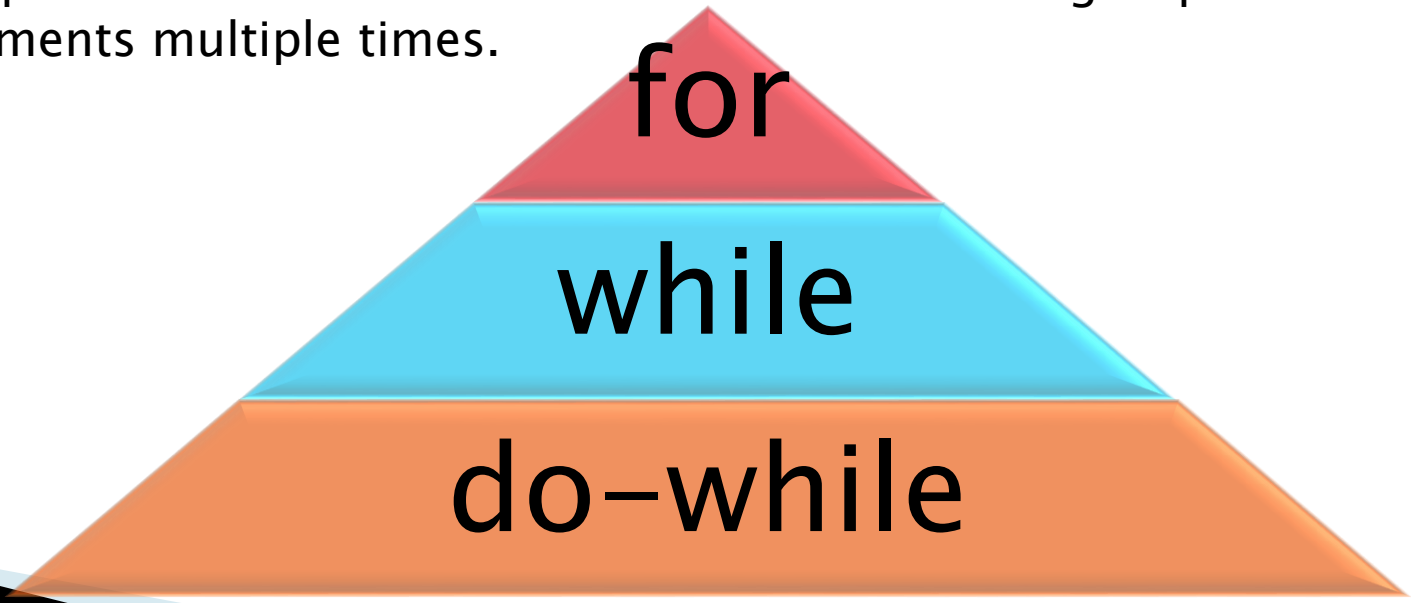
Ladder if:- An if statement can be followed by else if...else statement, which is very useful to test various conditions.

Syntax:-The syntax of an 'Ladder if' statement in C programming language is:

```
if(condition)
{
  statement(s);
 else if(condition)
   statement(s);
  else if(condition)
    statement(s);
  else
    statement(s);
}
```

Iterative Statements (Looping)

There are situations when a block of code needs to be executed several number of times. Generally, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on.
A loop statement allows to execute a statement or group of statements multiple times.



for loop

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

Syntax:-The syntax of a for loop in C programming language is:

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

Example

```
void main()
{
  int i;

  for(i=0;i<10;i++)
  {
      printf("%d ",i);
  }
 getch();
}
```

OUTPUT:

0 1 2 3 4 5 6 7 8 9

while loop

A while loop in C repeatedly executes a target statement as long as a given condition is true.

Syntax:-The syntax of a while loop in C programming language is:

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

Example

```
void main()
{
     int i=1;

  while(i<=10)
  {
     printf("%d\n",i);
     i++;
  }
 getch();
}
```

OUTPUT:

1 2 3 4 5 6 7 8 9 10

do-while loop

A do-while loop is similar to a while loop, except that it is guaranteed
to execute at least one time. It is also called exit control statement.

Syntax:-The syntax of a do-while loop in C programming language is:

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

Example

```
void main()
{
     int i=1;

 do
  {
     printf("%d\n",i);
     i++;
  }    while(i<=10);
 getch();
}
```

OUTPUT:

1 2 3 4 5 6 7 8 9 10

Jumping Statements

Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.

break:-break statement Terminates the loop or switch statement and transfers execution to the statement immediately following the loop or switch.

Syntax:- break;

continue:-continue statement Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.

Syntax:- continue;

goto:-goto statement Transfers control to the labeled statement.

Syntax:-goto label;

Thank You