* ***C PROGRAMING LANGUAGE***

**C language:-**

C is a procedural programming language initially developed by Dennis Ritchie in the year 1972 at Bell Laboratories of AT&T Labs. It was mainly developed as a system programming language to write the UNIX operating system

**Stdio.h:-**  stdio. h header file allows us to perform input and output operations in C. The functions like printf() and scanf() are used to display output and take input from a user.

**Conio.h:-** conio. h is a header file in which there are many built-in functions embedded in it they generally perform input/output on the console i.e., it is used to take input from the keyboard given by the user and display output on the screen.getch() and clrscr() functions are provided by the conio.h header file

**Printf():-**in C language, printf function is used to print output on the screen.  This function is a part of the C standard library “**stdio.h**” and it can allow formatting the output in numerous ways.

**Scanf():-**In C programming language, scanf is a function that stands for Scan Formatted String. It is used to read data from stdin (standard input stream i.e. usually keyboard) and then writes the result into the given arguments.

* It accepts character, string, and numeric data from the user using standard input.
* scanf also uses format specifiers like printf.

**Getch():-**getch() also reads a single character from the keyboard. But it does not use any buffer, so the entered character is immediately returned without waiting for the enter key.

**Clrscr():-**clrscr() is an abbreviation of the clear screen. It aims to clear the console screen. clrscr() is a library function located in the console input output header file <conio..h>

**Void main():-**Void main () is the entry point for execution in C program. The void is a keyword that represents function will not return anything but a void value. Main is the name of the function and () represents parameter list that can be passed to function in this case nothing is passed

**Format Specifier in C**

In C, a value can be a character type, integer type, float type, and so on. To display these values we have format specifiers used in printf function. These format specifiers start with the percentage symbol ‘%’. Some of the commonly used format specifiers are given below.

**%d -** for printing integers

**%f -** for printing floating-point numbers

**%c -** for printing characters

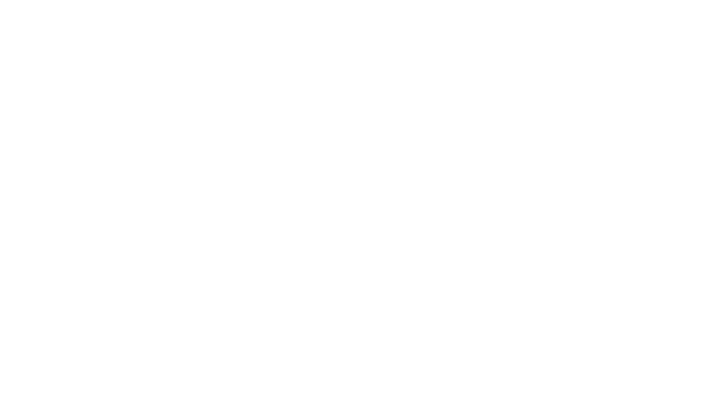
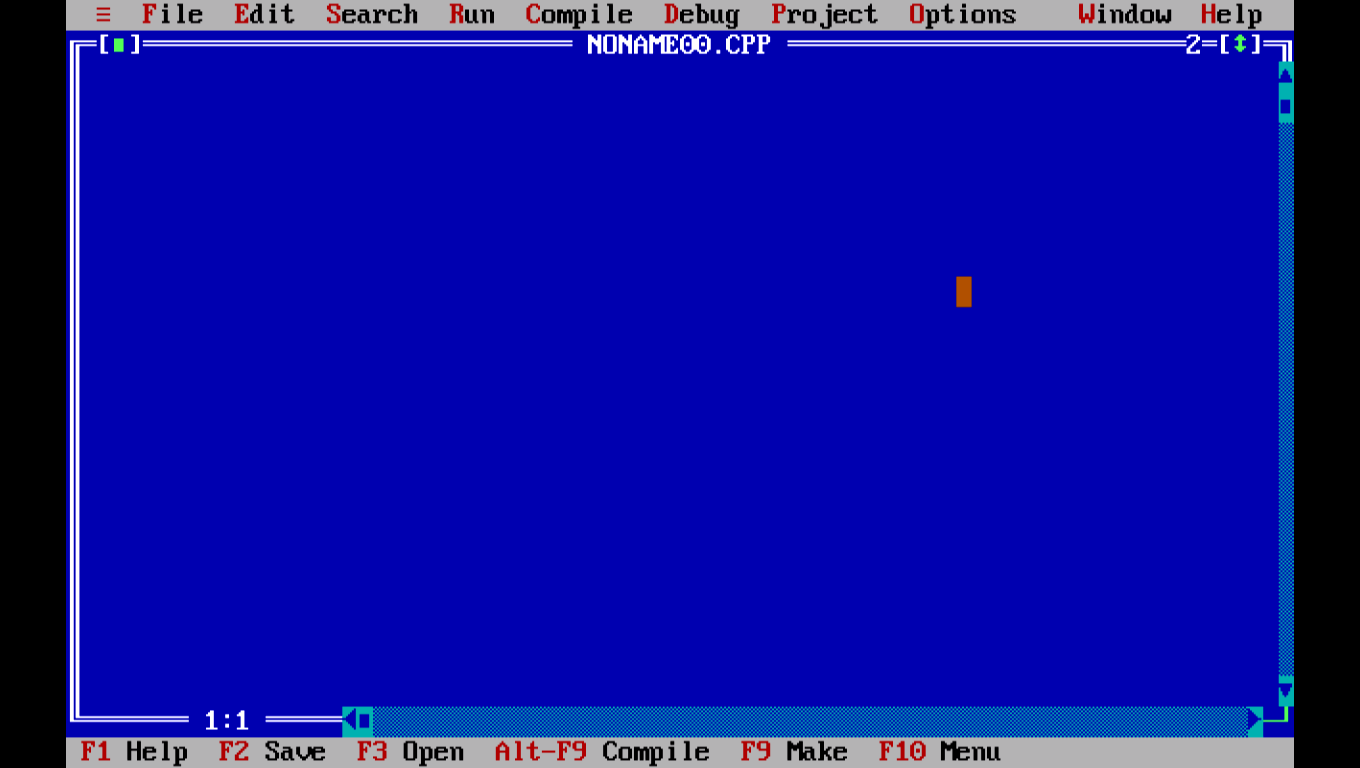
**%s -** for printing strings

**%p -** for printing memory addresses

**%x** - for printing hexadecimal values

BASIC STRUCTURE OF C SOFTWARE

TURBO C++ BASIC



**SHORTCUT KEY OF C PROGRAMING’S**

| **S No.** | **Shortcut keys** | **Action** |
| --- | --- | --- |
| 1. | F1 | For Help |
| 2. | F2 | Save |
| 3. | F3 | Open |
| 4. | F4 | Go to cursor |
| 5. | F5 | Zoom |
| 6. | F6 | Next |
| 7. | F7 | Trace into |
| 8. | F8 | Step over |
| 9. | F9 | Make |
| 10. | F10 | Menu |
| 11. | Alt + F1 | Go to the previous Topic |
| 12. | Alt + F3 | Close the opened file |
| 13. | Alt + F4 | Inspect |
| 14. | Alt + F5 | Open user screen/output screen |
| 15. | Alt + F7 | Go to the previous Error |
| 16. | Alt + F8 | Go to the next Error |
| 17. | Alt + F9 | Compile |
| 18. | Alt + 0 | Open a list of all available files |
| 19. | Alt + C | Open Compile menu option |
| 20. | Alt + D | Open Debug menu option |
| 21. | Alt + E | Open the Edit menu option |
| 22. | Alt + F | Open File menu option |
| 23. | Alt + H | Open the Help menu option |
| 24. | Alt + O | Open the option menu |
| 25. | Alt + P | Open Project menu option |
| 26. | Alt + R | Open Run menu option |
| 27. | Alt + S | Open Search menu option |
| 28. | Alt + W | Open Window menu option |
| 29. | Alt + X | Quit |
| 30. | Alt + Bksp | Undo |
| 31. | Alt + Bksp +  Shift | Redo |
| 32. | Alt + Enter | Minimize or Maximize |
| 33. | Ctrl + Delete | Delete selected code |
| 34. | Ctrl + F1 | Topic Search |
| 35. | Ctrl + F2 | Reset Opened Program |
| 36. | Ctrl + F3 | Call Stack |
| 37. | Ctrl + F4 | Modify/Evaluate Expression |
| 38. | Ctrl + F5 | Size/Move |
| 39. | Ctrl + F7 | Add Watch |
| 40. | Ctrl + F8 | Toggle between breakpoints |
| 41. | Ctrl + F9  ‘or’  Alt + R + Enter | Run code |
| 42. | Ctrl + L | Search selected String |
| 43. | Ctrl + N | Add New Line |
| 44. | Ctrl + S | Save |
| 45. | Ctrl + Y | Delete Line |
| 46. | Ctrl + Ins | Copy |
| 47. | Ctrl + K + B | Select Starting Points |
| 48. | Ctrl + K + K | Select Ending Points |
| 49. | Ctrl + K + H | Hide Grey line of Selection |
| 50. | Windows | Exit |
| 51. | Shift + Delete | Cut |
| 52. | Shift + Ins  ‘or’  Ctrl + K + C | Paste |
| 53. | Shift + F1 | Open Turbo Help Index |

* **BASIC C PROGRAMING CODE :**
* **ADDITION OF 2 NUMBERS:**

#include <stdio.h>

#include<conio.h>

Void main()

{

int number1, number2, sum;

clrscr();

printf("Enter two integers: ");

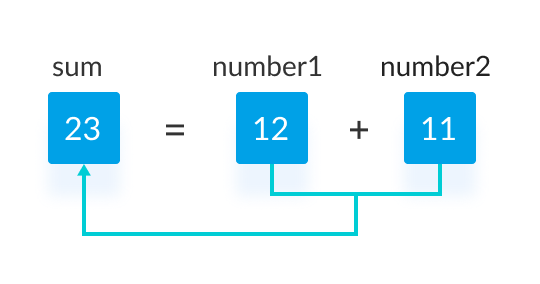
scanf("%d %d", &number1, &number2);

sum = number1 + number2;

printf("%d + %d = %d", number1, number2, sum);

getch();

}



* **SUBTRACTION OF 2 NUMBER’S:**

include <stdio.h>

Void main()

{

int number1, number2, subtract;

printf("Enter two integers: ");

scanf("%d %d", &number1, &number2);

subtract = number1 - number2;

printf("%d - %d = %d", number1, number2, subtract);

}

* **Program to Multiplication of two numbers**

#include<stdio.h>

#include<conio.h>

void main()

{

int num1,num2,product;

printf("Enter two numbers:");

scanf("%d %d",&num1,&num2);

product=num1\*num2;

printf("Product of two numbers: %d",product);

getch(); }

* **Program to division of two numbers**

#include<stdio.h>

#include<conio.h>

Void main()

{

int num1,num2,quotient;

printf("Enter two numbers:");

scanf("%d %d",&num1,&num2);

quotient=num1/num2;

printf("Quotient: %d",quotient);

getch(); }

* **AREA OF SQUARE:**

#include<stdio.h>

#include<conio.h>

Void main()

{

int side,area;

clrscr();

printf("Enter a side of square:");

scanf("%d",&side);

area=side \* side;

printf("\nArea of Square: %d",area);

getch();

}

# **C Program to find area of Rectangle**

#include<stdio.h>

#include<conio.h>

Void main()

{

int length,breadth, area;

clrscr();

printf("Enter length and breadth of rectangle:");

scanf("%d %d",&length,&breadth);

area=length \* breadth;

printf("\nArea of Rectangle: %d",area);

getch();

}

* **Simple interest:**

#include<stdio.h>

#include<conio.h>

Void main()

{

float p,r,t,si;

clrscr();

printf(“enter the p and t and r=”)

scanf("%f %f %f",&p,&t,&r);

si=(p\*t\*r)/100;

printf("%f",si);

getch();

}

# **C Program to convert FAHRENHIET TO CELSIUS**

#include<stdio.h>

#include<conio.h>

Void main()

{

float f,c;

clrscr();

printf("Enter celsius:");

scanf("%f",&c);

f=c\*1.8+32;

printf("Fahrenhiet:%f",f);

getch();

}

* AREA OF TRIANGLE:

#**include** <stdio.h>

#include<conio.h>

**void** main()

{

**float** base, height, area;

clrscr();

printf("Enter base of the triangle: ");

scanf("%f", &base);

printf("Enter height of the triangle: ");

scanf("%f", &height);

area = (base \* height) / 2;

printf("Area of the triangle = %.2f sq. units", area);

getch();

}

* AREA OF CIRCLE:

#include<stdio.h>

#include<conio.h>

Void main()

{

const float pi=3.14;

int radius;

float area;

printf("Enter Radius:");

scanf("%d",&radius);

area=pi \* (radius \* radius);

printf("Area of Circle: %f",area);

getch();

}

* **IF CONDITION**
* Even or odd:

#include <stdio.h>

#include<conio.h>

Void main()

{

int num;

printf("Enter an integer: ");

scanf("%d", &num);

// true if num is perfectly divisible by 2

if(num % 2 = = 0)

printf("%d is even.", num);

else

printf("%d is odd.", num);

getch();

}

* **LEAP YEAR OR NOT:**

#include <stdio.h>

#include<conio.h>

Void main()

{

int year;

printf("Enter a year: ");

scanf("%d", &year);

// leap year if perfectly divisible by 400

if (year % 400 == 0) {

printf("%d is a leap year.", year);

}

// not a leap year if divisible by 100

// but not divisible by 400

else if (year % 100 == 0) {

printf("%d is not a leap year.", year);

}

// leap year if not divisible by 100

// but divisible by 4

else if (year % 4 == 0) {

printf("%d is a leap year.", year);

}

// all other years are not leap years

else {

printf("%d is not a leap year.", year);

}

getch();

}

* **C program to find maximum between three numbers**

#include <stdio.h>

#include<conio.h>

void main()

{

float n1, n2, n3;

printf("Enter three different numbers: ");

scanf("%f %f %f", &n1, &n2, &n3);

// if n1 is greater than both n2 and n3, n1 is the largest

if (n1 >= n2 && n1 >= n3)

printf("%f is the largest number.", n1);

// if n2 is greater than both n1 and n3, n2 is the largest

if (n2 >= n1 && n2 >= n3)

printf("%f is the largest number.", n2);

// if n3 is greater than both n1 and n2, n3 is the largest

if (n3 >= n1 && n3 >= n2)

printf("%f is the largest number.", n3);

getch();

}

* **Student grade calculation:**

#include<stdio.h>

#include<conio.h>

Void main()

{

int num;

printf("Enter your mark ");

scanf("%d",&num);

printf(" You entered %d", num); // printing outputs

if(num >= 80){

printf(" You got A grade"); // printing outputs

}

else if ( num >=60){ // Note the space between else & if

printf(" You got B grade");

}

else if ( num >=40)

{

printf(" You got C grade");

}

else if ( num < 40){

printf(" You Failed in this exam");}

getch(); }

* **Metre to kilometre convert:**

#include <stdio.h>

#include<conio.h>

Void main

{

int m;

float km;

printf("Enter Meters:");

scanf("%d",&m);

km=m/1000.00;

printf("Kilometers: %f",km);

getch();

}

* C program to calculate gross salary of an employee

#include <stdio.h>

#include<conio.h>

Void main()

{

float basic, gross, da, hra;

printf(“Enter basic salary of an employee: “);

scanf(“%f”, &basic);

if(basic <= 10000)

{

da = basic \* 0.8;

hra = basic \* 0.2;

}

else if(basic <= 20000)

{

da = basic \* 0.9;

hra = basic \* 0.25;

}

else

{

da = basic \* 0.95;

hra = basic \* 0.3;

}

/\* Calculate gross salary \*/

gross = basic + hra + da;

printf(“GROSS SALARY OF EMPLOYEE = %f”, gross);

getch();

}

* **AMSTRONG NUMBER:**

#include <math.h>

#include <stdio.h>

#include<conio.h>

void main()

{

int num, originalNum, remainder, n = 0;

float result = 0.0;

printf(“Enter an integer: “);

scanf(“%d”, &num);

originalNum = num;

// store the number of digits of num in n

for (originalNum = num; originalNum != 0; ++n) {

originalNum /= 10;

}

for (originalNum = num; originalNum != 0; originalNum /= 10) {

remainder = originalNum % 10;

// store the sum of the power of individual digits in result

result += pow(remainder, n);

}

// if num is equal to result, the number is an Armstrong number

if ((int)result == num)

printf(“%d is an Armstrong number.”, num);

else

printf(“%d is not an Armstrong number.”, num);

getch();

}

* **Palindrome number:**

#include<stdio.h>

#include<conio.h>

Void main()

{

int n,r,sum=0,temp;

printf(“enter the number=”);

scanf(“%d”,&n);

temp=n;

while(n>0)

{

r=n%10;

sum=(sum\*10)+r;

n=n/10;

}

if(temp==sum)

printf(“palindrome number “);

else

printf(“not palindrome”);

getch(); }

* **Reverse number:**

#include<stdio.h>

#include<conio.h>

void main()

{

int num, rev = 0, rem;

printf(“Enter the number to reverse: “);

scanf(“%d”, &num);

while (num != 0){

rem = num % 10;

rev = rev \* 10 + rem;

num = num/10;

}

printf(“The reversed number is: %d”, rev);

getch();

}

**Sum of 1 to 10 even number:**

#include <stdio.h>

include <conio.h>

Void main()

{

int i,n=2,s=0;

for(i=1; i<=10; i=i+1)

{

printf("%d\n",n);

s=s+n;

n=n+2;

}

printf("Sum of 1st 10 even numbers = %d",s);

getch();

}

* **SUM OF 1 TO 10 0DD NUMBER:**

#include <stdio.h>

#include<conio.h>

Void main()

{

int i, n, sum=0;

printf("Enter upper limit: ");

scanf("%d", &n);

/\* Find the sum of all odd number \*/

for(i=1; i<=n; i+=2)

{

sum += i;

}

printf("Sum of odd numbers = %d", sum);

getch();

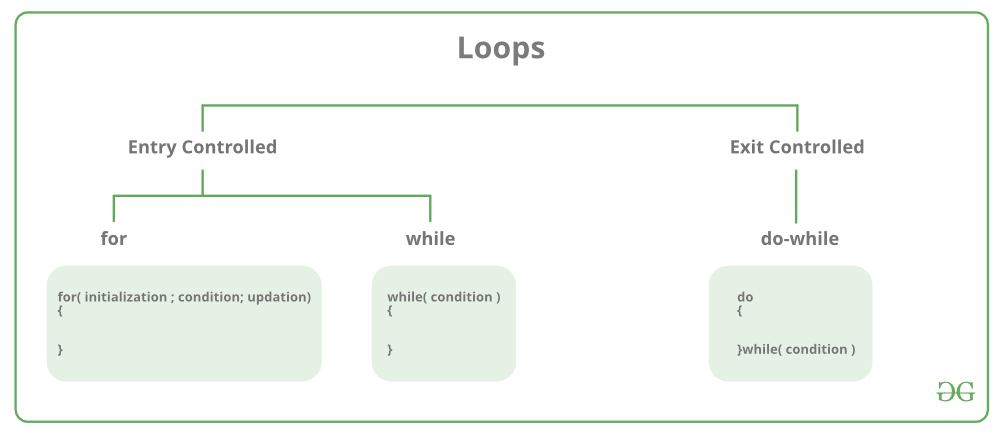
}

* **LOOP IN C PROGRAMME**

Loops in programming are used to repeat a block of code until the specified condition is met. A loop statement allows programmers to execute a statement or group of statements multiple times without repetition of code.

* **here are mainly two types of loops in C Programming:**

1. **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.
2. **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.



| **Loop Type** | **Description** |
| --- | --- |
| for loop | first Initializes, then condition check, then executes the body and at last, the update is done. |
| while loop | first Initializes, then condition checks, and then executes the body, and updating can be inside the body. |
| do-while loop | do-while first executes the body and then the condition check is done. |

* **for Loop**

for loop in C programming is a  repetition control structure that allows programmers to write a loop that will be executed a specific number of times. for loop enables programmers to perform n number of steps together in a single line.

In for loop, a loop variable is used to control the loop. Firstly we initialize the loop variable with some value, then check its test condition. If the statement is true then control will move to the body and the body of for loop will be executed. Steps will be repeated till the exit condition becomes true. If the test condition will be false then it will stop.

* **Initialization Expression:**In this expression, we assign aloop variable or loop counter to some value. for example: int i=1;
* **Test Expression:**In this expression, test conditions are performed. If the condition evaluates to true then the loop body will be executed and then an update of the loop variable is done. If the test expression becomes false then the control will exit from the loop. for example, i<=9;
* **Update Expression:**After execution of the loop body loop variable is updated by some value it could be incremented, decremented, multiplied, or divided by any value.

## While Loop

While loop does not depend upon the number of iterations. In for loop the number of iterations was previously known to us but in the While loop, the execution is terminated on the basis of the test condition. If the test condition will become false then it will break from the while loop else body will be executed.

## do-while Loop

The do-while loop is similar to a while loop but the only difference lies in the do-while loop test condition which is tested at the end of the body. In the do-while loop, the loop body will **execute at least once** irrespective of the test condition.