Constants in C

A constant is a value or variable that can't be changed in the program, for example, 10, 20, 'a', 3.4, "c programming" etc.

There are different types of constants in C programming.

2 ways to define constant in C

There are two ways to define constant in C programming.

1. const keyword

2. #define pre-processor

1) C const keyword

The const keyword is used to define constant in C programming.

1. const float PI=3.14;

Now, the value of PI variable can't be changed.

1. #include <stdio.h>

2. #include <conio.h>

3. void main(){

4. const float PI=3.14;

5. clrscr();

6. printf("The value of PI is: %f",PI);

7. getch();

8. }

Output:

The value of PI is: 3.140000

If you try to change the the value of PI, it will render compile time error.

1. #include <stdio.h>

2. #include <conio.h>

3. void main(){

4. const float PI=3.14;

5. clrscr();

6. PI=4.5;

7. printf("The value of PI is: %f",PI);

8. getch();

9. }

Output:

Compile Time Error: Cannot modify a const object

2) C #define preprocessor

The #define preprocessor is also used to define constant. We will learn about #define preprocessor directive later.

C if-else Statement

The if statement in C language is used to perform an operation on the basis of condition. By using if-else statement, you can perform operation either condition is true or false.

There are many ways to use if statement in C language:

o If statement

o If-else statement

o If else-if ladder

o Nested if

If Statement

The single if statement in C language is used to execute the code if the condition is true. The syntax of if statement is given below:

1. if(expression){

2. //code to be executed

3. }

Flowchart of if statement in C

Let's see a simple example of c language if statement.

1. #include<stdio.h>

2. #include<conio.h>

3. void main(){

4. int number=0;

5. clrscr();

6.

7. printf("enter a number:");

8. scanf("%d",&number);

9.

10. if(number%2==0){

11. printf("%d is even number",number);

12. }

13.

14. getch();

15. }

Output

enter a number:4

4 is even number

enter a number:5

If-else Statement

The if-else statement in C language is used to execute the code if condition is true or false. The syntax of if-else statement is given below:

1. if(expression){

2. //code to be executed if condition is true

3. }else{

4. //code to be executed if condition is false

5. }

Flowchart of an if-else statement in C

Let's see the simple example of even and odd number using if-else statement in C language.

1. #include<stdio.h>

2. #include<conio.h>

3. void main(){

4. int number=0;

5. clrscr();

6.

7. printf("enter a number:");

8. scanf("%d",&number);

9.

10. if(number%2==0){

11. printf("%d is even number",number);

12. }

13. else{

14. printf("%d is odd number",number);

15. }

16. getch();

17. }

Output

enter a number:4

4 is even number

enter a number:5

5 is odd number

If else-if ladder Statement

The if else-if statement is used to execute one code from multiple conditions. The syntax of if else-if statement is given below:

1. if(condition1){

2. //code to be executed if condition1 is true

3. }else if(condition2){

4. //code to be executed if condition2 is true

5. }

6. else if(condition3){

7. //code to be executed if condition3 is true

8. }

9. ...

10. else{

11. //code to be executed if all the conditions are false

12. }

Flowchart of else-if ladder statement in C

The example of if-else-if statement in C language is given below.

1. #include<stdio.h>

2. #include<conio.h>

3. void main(){

4. int number=0;

5. clrscr();

6.

7. printf("enter a number:");

8. scanf("%d",&number);

9.

10. if(number==10){

11. printf("number is equals to 10");

12. }

13. else if(number==50){

14. printf("number is equal to 50");

15. }

16. else if(number==100){

17. printf("number is equal to 100");

18. }

19. else{

20. printf("number is not equal to 10, 50 or 100");

21. }

22. getch();

23. }

Output

enter a number:4

number is not equal to 10, 50 or 100

enter a number:50

number is equal to 50

C Switch Statement

The switch statement in C language is used to execute the code from multiple conditions. It is like if else-if ladder statement.

The syntax of switch statement in c language is given below:

1. switch(expression){

2. case value1:

3. //code to be executed;

4. break; //optional

5. case value2:

6. //code to be executed;

7. break; //optional

8. ......

9.

10. default:

11. code to be executed if all cases are not matched;

12. }

Rules for switch statement in C language

1) The switch expression must be of integer or character type.

2) The case value must be integer or character constant.

3) The case value can be used only inside the switch statement.

4) The break statement in switch case is not must. It is optional. If there is no break statement found in switch case, all the cases will be executed after matching the case value. It is known as fall through state of C switch statement.

Let's try to understand it by the examples. We are assuming there are following variables.

1. int x,y,z;

2. char a,b;

3. float f;

Valid Switch Invalid Switch Valid Case Invalid Case

switch(x) switch(f) case 3; case 2.5;

switch(x>y) switch(x+2.5) case 'a'; case x;

switch(a+b-2) case 1+2; case x+2;

switch(func(x,y)) case 'x'>'y'; case 1,2,3;

Flowchart of switch statement in C

Let's see a simple example of c language switch statement.

1. #include<stdio.h>

2. #include<conio.h>

3. void main(){

4. int number=0;

5. clrscr();

6.

7. printf("enter a number:");

8. scanf("%d",&number);

9.

10. switch(number){

11. case 10:

12. printf("number is equals to 10");

13. break;

14. case 50:

15. printf("number is equal to 50");

16. break;

17. case 100:

18. printf("number is equal to 100");

19. break;

20. default:

21. printf("number is not equal to 10, 50 or 100");

22. }

23. getch();

24. }

Output

enter a number:4

number is not equal to 10, 50 or 100

enter a number:50

number is equal to 50

C Switch statement is fall-through

In C language, switch statement is fall through, it means if you don't use break statement in switch case, all the case after matching case will be executed.

Let's try to understand the fall through state of switch statement by the example given below.

1. #include<stdio.h>

2. #include<conio.h>

3. void main(){

4. int number=0;

5. clrscr();

6.

7. printf("enter a number:");

8. scanf("%d",&number);

9.

10. switch(number){

11. case 10:

12. printf("number is equals to 10\n");

13. case 50:

14. printf("number is equal to 50\n");

15. case 100:

16. printf("number is equal to 100\n");

17. default:

18. printf("number is not equal to 10, 50 or 100");

19. }

20. getch();

21. }

Output

enter a number:10

number is equals to 10

number is equals to 50

number is equals to 100

number is not equal to 10, 50 or 100

enter a number:50

number is equal to 50

number is equals to 100

number is not equal to 10, 50 or 100

C Loops

The loops in C language are used to execute a block of code or a part of the program several times.

In other words, it iterates a code or group of code many times.

Why use loops in C language?

Suppose that you have to print a table of 2, then you need to write 10 lines of code.

By using the loop statement, you can do it by 2 or 3 lines of code only.

Advantage of loops in C

1) It saves code.

2) It helps to traverse the elements of array (which is covered in next pages).

Types of C Loops

There are three types of loops in C language that is given below:

1. do while

2. while

3. for

do-while loop in C

It iterates the code until a condition is false. Here, the condition is given after the code. So at least once, code is executed whether the condition is true or false.

It is better if you have to execute the code at least once.

The syntax of do-while loop in c language is given below:

1. do{

2. //code to be executed

3. }while(condition);

while loop in C

Like, do while it iterates the code until a condition is false. Here, the condition is given before the code. So code may be executed 0 or more times.

It is better if the number of iteration is not known by the user.

The syntax of while loop in c language is given below:

1. while(condition){

2. //code to be executed

3. }

for loop in C

Like while it iterates the code until a condition is false. Here, initialization, condition and increment/decrement is given before the code.

So code may be executed 0 or more times.

It is good if the number of iteration is known by the user.

The syntax of for loop in c language is given below:

1. for(initialization;condition;incr/decr){

2. //code to be executed

3. }

Infinitive while loop in C

If you pass 1 as an expression in while loop, it will run an infinite number of times

Infinitive for loop in C

If you don't initialize any variable, check condition and increment or decrement variable in for loop, it is known as infinitive for a loop.

In other words, if you place 2 semicolons in for loop, it is known as infinitive for a loop.

1. for(;;){

2. printf("infinitive for loop example by javatpoint");

3. }

C break statement

The break statement in C language is used to break the execution of loop (while, do-while and for) and switch case.

In the case of inner loops, it terminates the control of inner loop only.

There can be two usages of C break keyword:

1. With switch case

2. With loop

C continue statement

The continue statement in C language is used to continue the execution of loop (while, do while and for). It is used with if condition within the loop.

In the case of inner loops, it continues the control of inner loop only.

C goto statement

The goto statement is known as jump statement in C language. It is used to unconditionally jump to other labels. It transfers control to other parts of the program.

It is rarely used today because it makes the program less readable and complex.

Type Casting in C

Typecasting allows us to convert one data type into others. In C language, we use cast operator for typecasting which is denoted by (type).

Syntax:

1. (type)value;

Note: It is always recommended to convert the lower value to higher for avoiding data loss.

Without Type Casting:

1. int f= 9/4;

2. printf("f : %d\n", f );//Output: 2

With Type Casting:

1. float f=(float) 9/4;

2. printf("f : %f\n", f );//Output: 2.250000

Type Casting example

Let's see a simple example to cast int value into float.

1. #include <stdio.h>

2. #include <conio.h>

3. void main(){

4. clrscr();

5.

6. float f= (float)9/4;

7. printf("f : %f\n", f );

8.

9. getch();

10. }

Output:

f : 2.250000