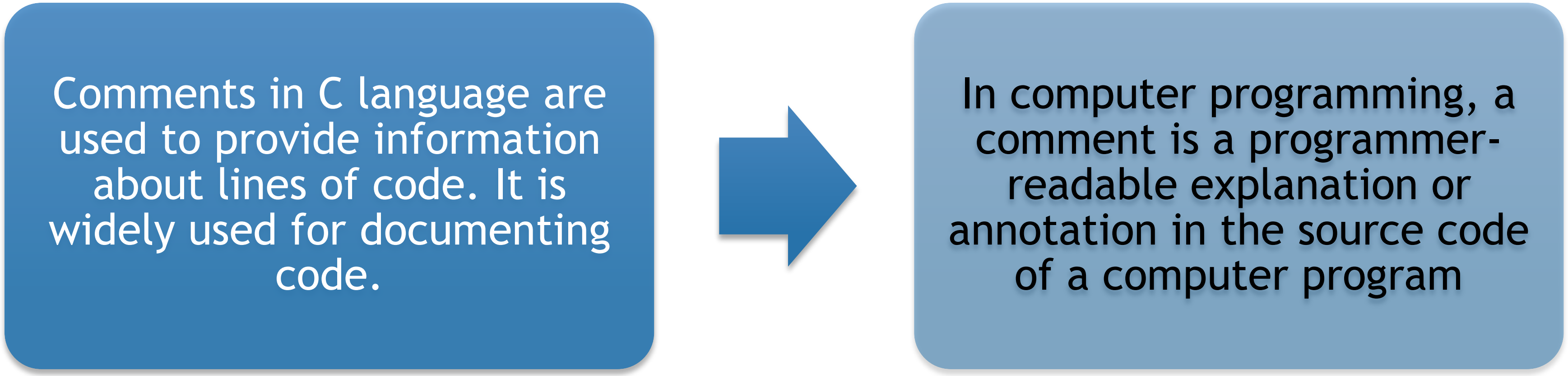
Aspirevision Tech™
Education Private Limited

PROGRAMMING WITH C

BY SHIVAM KUMAR

www.mtaeducation.in

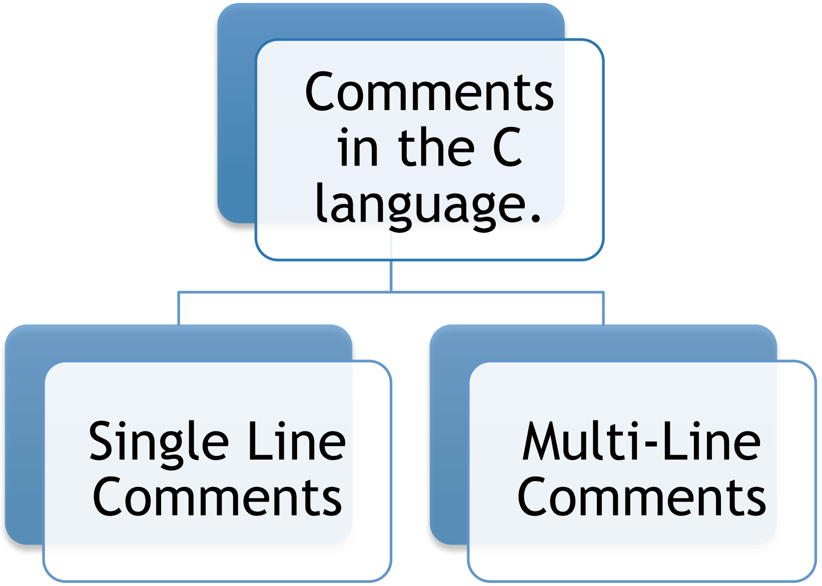
COMMENTS IN C



Comments in C language are used to provide information about lines of code. It is widely used for documenting code.

In computer programming, a comment is a programmer-readable explanation or annotation in the source code of a computer program

COMMENTS IN C


Comments in the C language.

Single Line Comments

Multi-Line Comments

SINGLE LINE COMMENTS

Single line comments are represented by double slash \\.

```
#include<stdio.h>
int main(){
    //printing information
     printf("Hello C");
return 0;
}
```

SINGLE LINE COMMENTS

Even you can place the comment after the statement.

```
#include<stdio.h>
int main(){
    printf("Hello C");//printing information
return 0;
}
```

MULT LINE COMMENTS

Multi-Line comments are represented by slash asterisk * ... *\. It can occupy many lines of code, but it can't be nested.

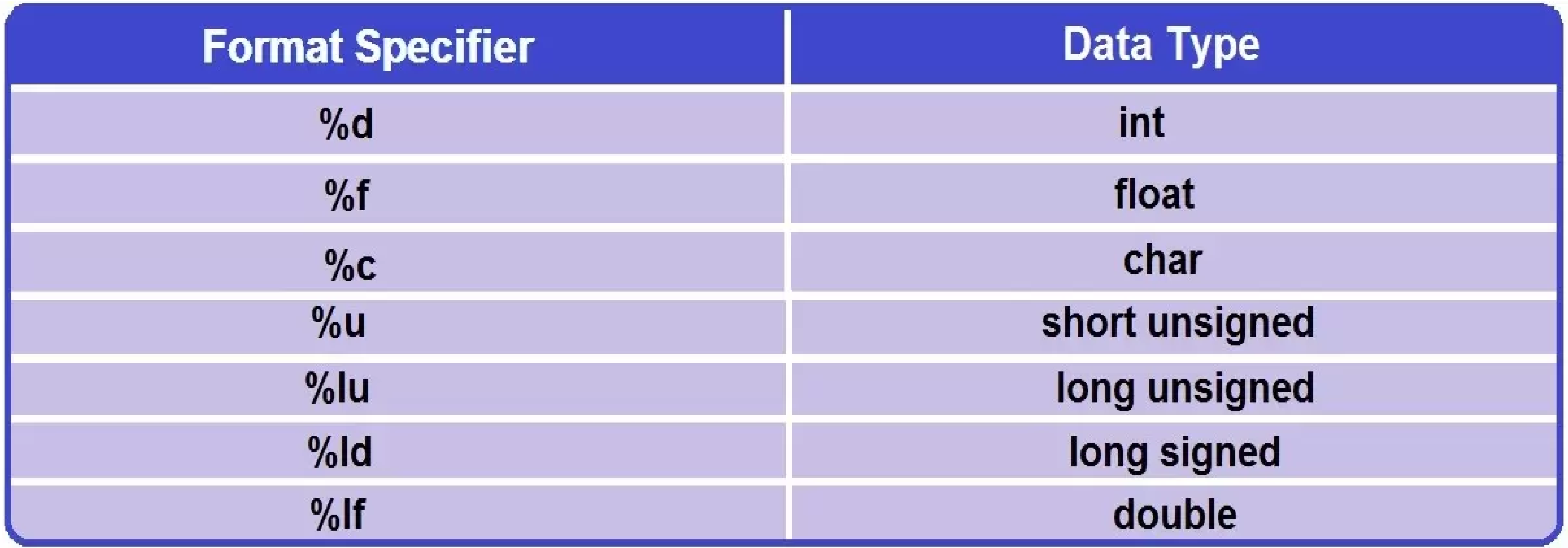
Syntax:

```
/*
code
to be commented
*/
```

```
#include<stdio.h>
int main(){
     /*printing information
        Multi-Line Comment*/
        printf("Hello C");
return 0;
}
```

C Format Specifier

The Format specifier is a string used in the formatted input and output functions. The format string determines the format of the input and output. The format string always starts with a '%' character.

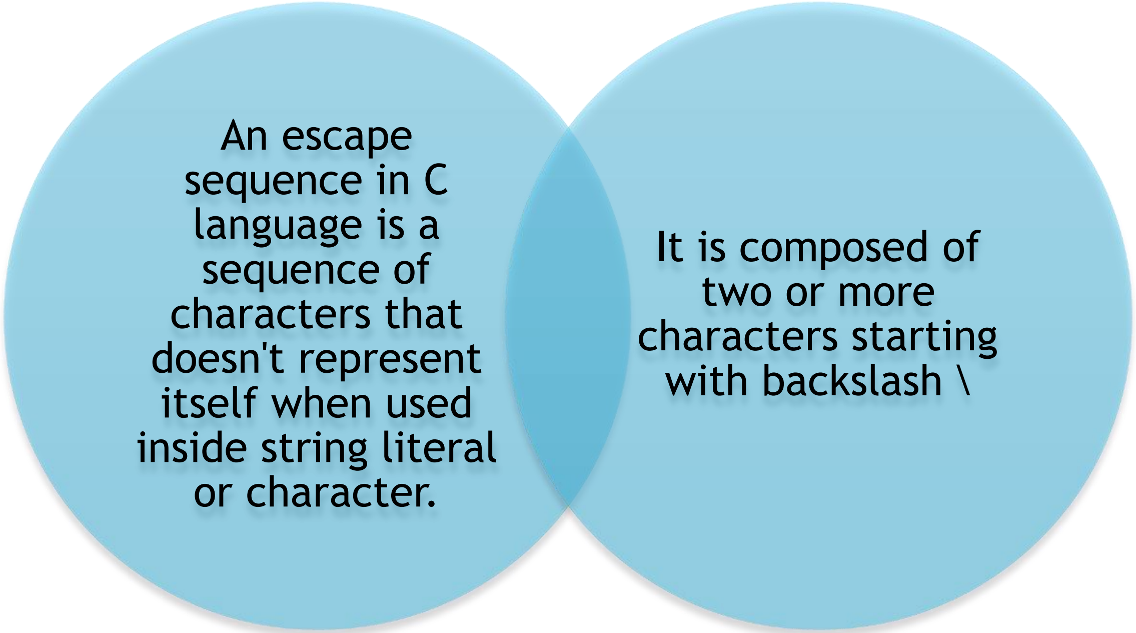


Format Specifier	Data Type
%d	int
%f	float
%c	char
%u	short unsigned
%lu	long unsigned
%ld	long signed
%lf	double

C Format Specifier

```
int main()
{
    int b=6;
    int c=8;
    printf("Value of b is:%d", b);
    printf("\nValue of c is:%d",c);
   return 0;
}
```

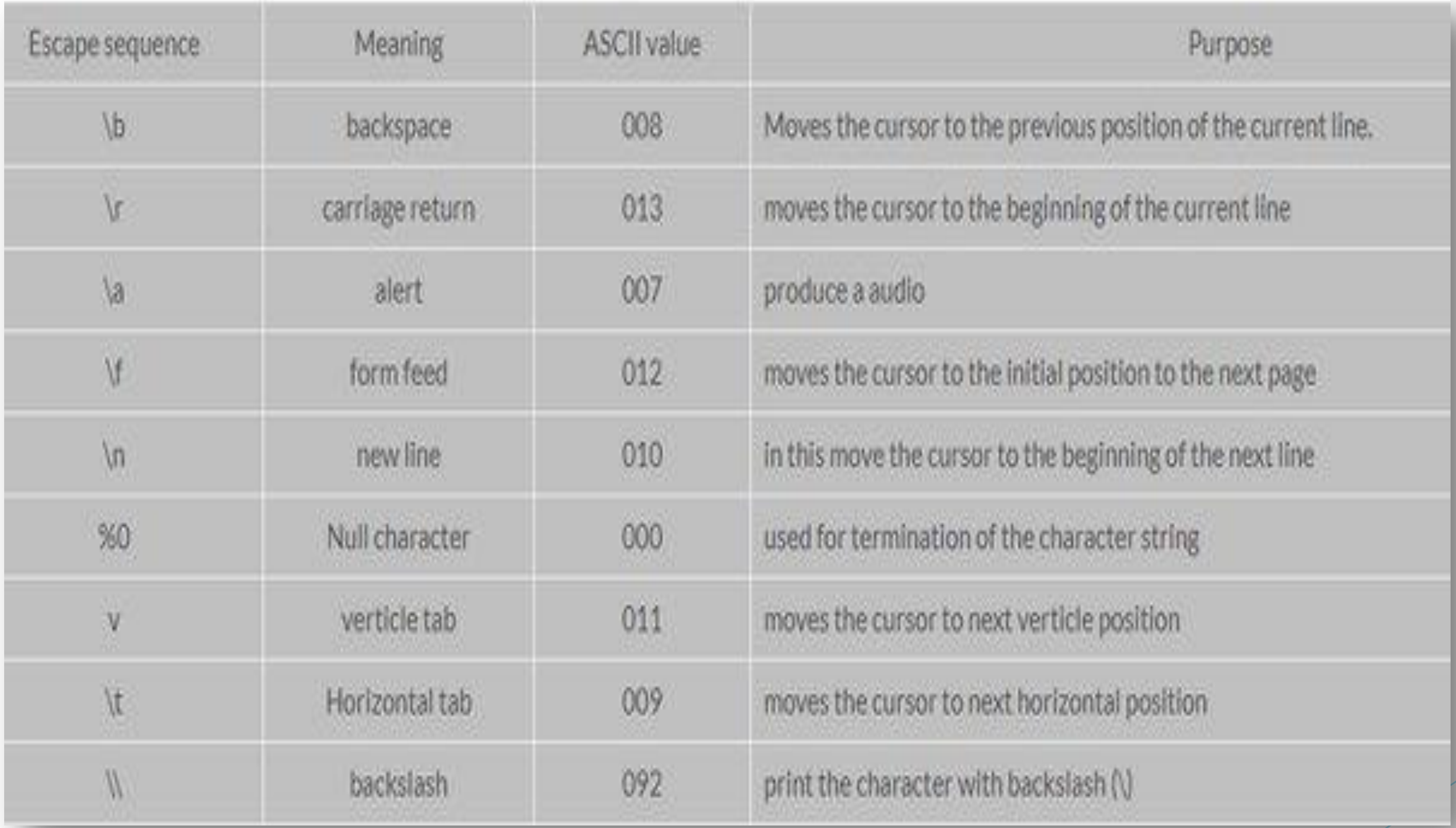
Escape Sequence in C



An escape sequence in C language is a sequence of characters that doesn't represent itself when used inside string literal or character.

It is composed of two or more characters starting with backslash \

Escape Sequence in C

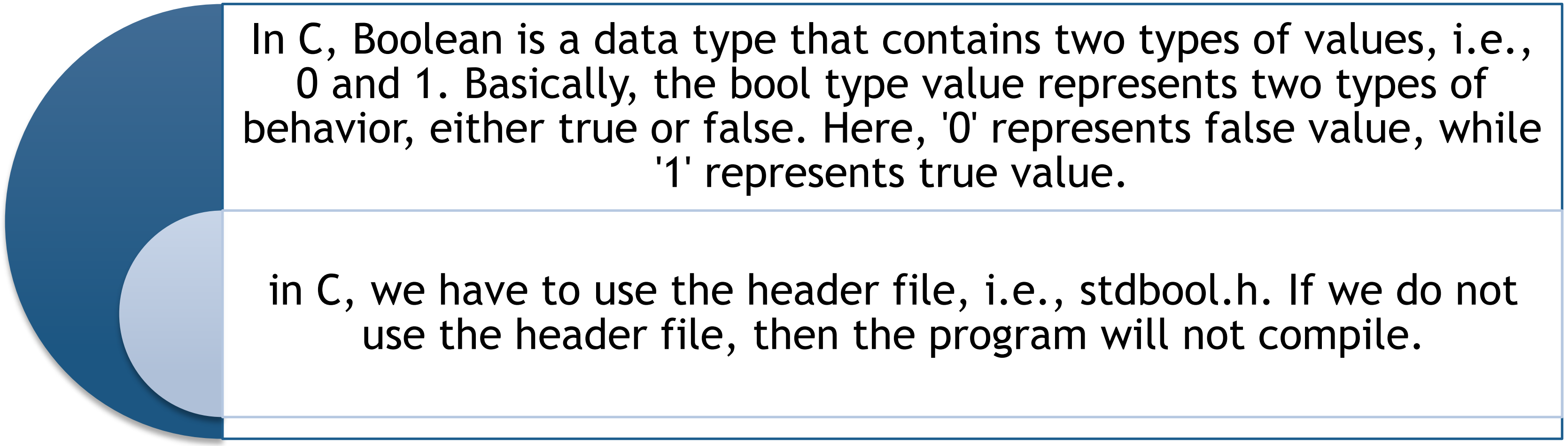


Escape sequence	Meaning	ASCII value	Purpose
\b	backspace	008	Moves the cursor to the previous position of the current line.
\r	carriage return	013	moves the cursor to the beginning of the current line
\a	alert	007	produce a audio
\f	form feed	012	moves the cursor to the initial position to the next page
\n	new line	010	in this move the cursor to the beginning of the next line
%0	Null character	000	used for termination of the character string
v	verticle tab	011	moves the cursor to next verticle position
\t	Horizontal tab	009	moves the cursor to next horizontal position
\\	backslash	092	print the character with backslash (\)

Escape Sequence in C

```
#include<stdio.h>
int main(){
     int number=50;
    printf("You\nare\nlearning\n\'c\' language\n\"Do you know C language\"");
return 0;
}
```

C Boolean



In C, Boolean is a data type that contains two types of values, i.e., 0 and 1. Basically, the bool type value represents two types of behavior, either true or false. Here, '0' represents false value, while '1' represents true value.

in C, we have to use the header file, i.e., stdbool.h. If we do not use the header file, then the program will not compile.

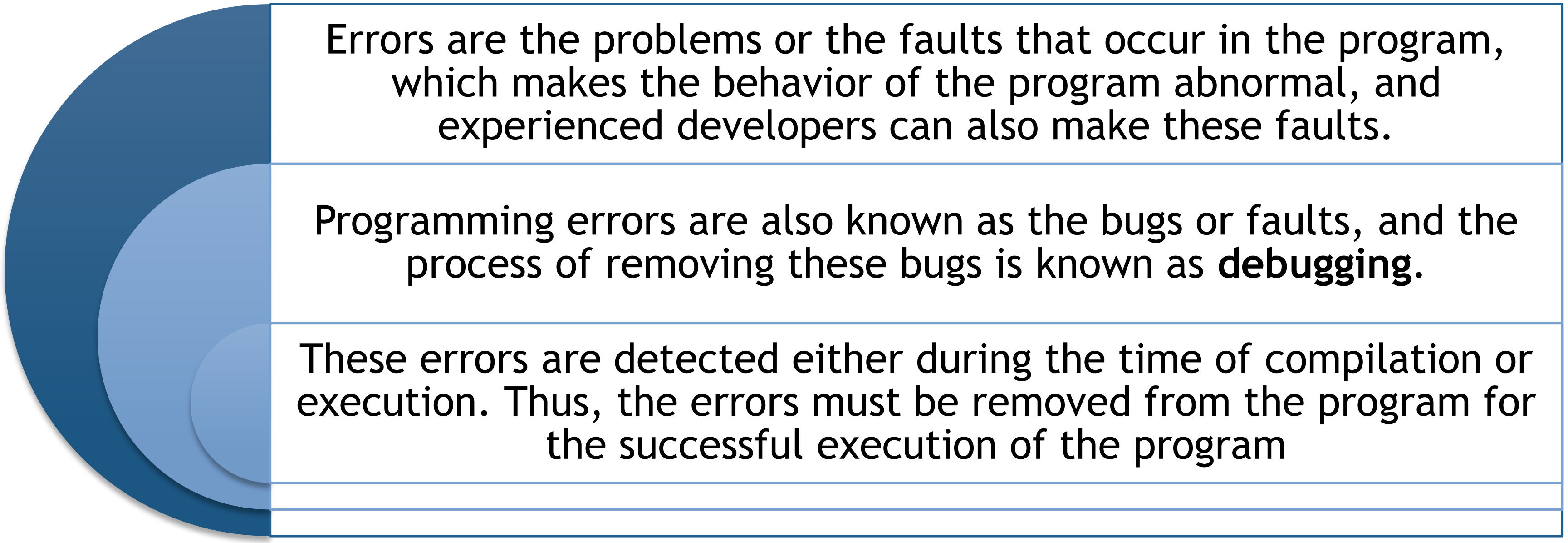
Syntax:

```
bool variable_name;
```

C Boolean

```
#include <stdio.h>
#include<stdbool.h>
int main()
{
     bool x=false; // variable initialization.
     if(x==true) // conditional statements
     {
         printf("The value of x is true");
     }
    else
    {
        printf("The value of x is FALSE");
    }
   return 0;
}
```

Programming Errors in C

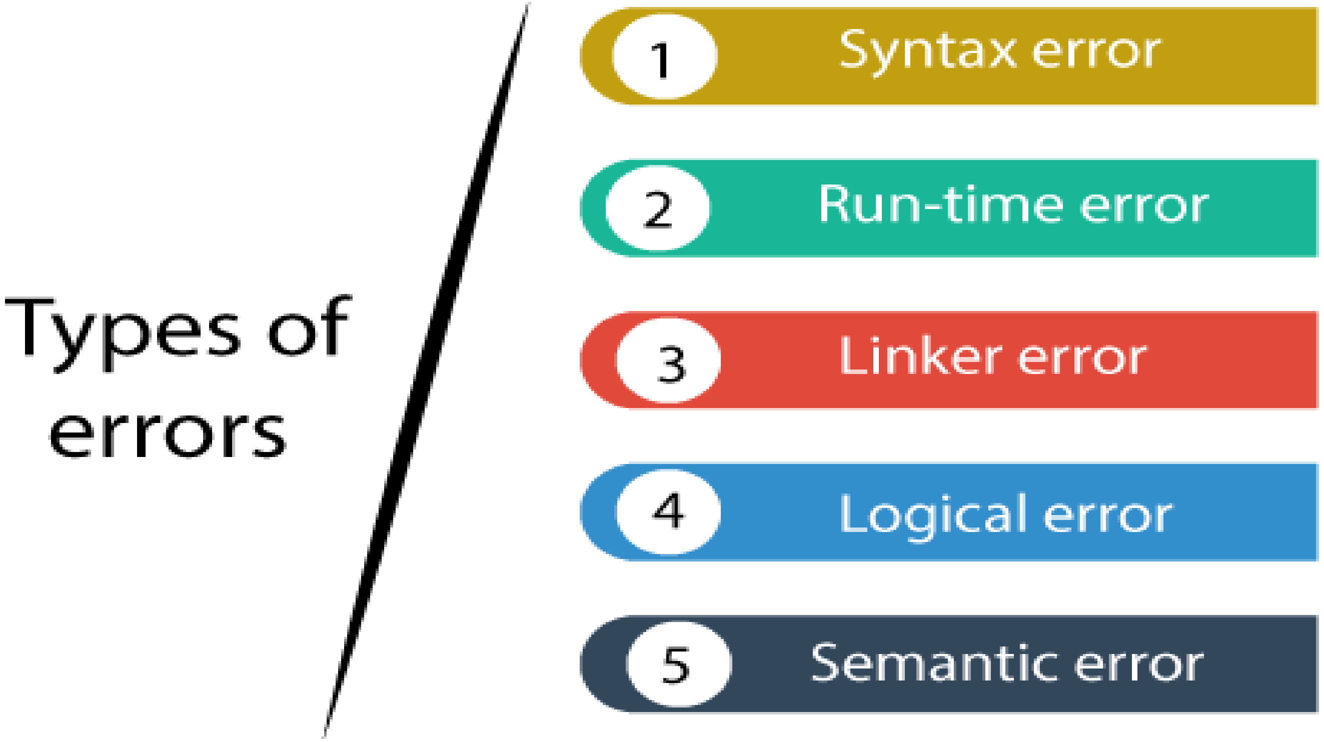


Errors are the problems or the faults that occur in the program, which makes the behavior of the program abnormal, and experienced developers can also make these faults.

Programming errors are also known as the bugs or faults, and the process of removing these bugs is known as **debugging**.

These errors are detected either during the time of compilation or execution. Thus, the errors must be removed from the program for the successful execution of the program

Programming Errors in C



Types of errors

1 Syntax error

2 Run-time error

3 Linker error

4 Logical error

5 Semantic error

Syntax error

```
#include <stdio.h>
int main()
{
     a = 10;
    printf("The value of a is : %d", a);
    return 0;
}
```

```
#include <stdio.h>
int main()
{
     int a=2;
     if(.)  // syntax error

     printf("a is greater than 1");
    return 0;
}
```

Run-time error

```
#include <stdio.h>
int main()
{
    int a=2;
    int b=2/0;
    printf("The value of b is : %d", b);
    return 0;
}
```

Linker error

```
#include <stdio.h>
int Main()
{
    int a=78;
    printf("The value of a is : %d", a);
    return 0;
}
```

Logical error

```
#include <stdio.h>
int main()
{
      int sum=0; // variable initialization
      int k=1;
     for(int i=1;i<=10;i++); // logical error, as we put the semicolon after loop
     {
          sum=sum+k;
          k++;
     }
    printf("The  value of sum is %d", sum);
    return 0;
}
```

Semantic error

```
#include <stdio.h>
int main()
{
int a,b,c;
a=2;
b=3;
c=1;
a+b=c; // semantic error
return 0;
}
```