

9/10/2018

Errors in C/C++ - GeeksforGeeks

```
void main()  
{  
  int a, b, c;  
  a + b = c; //semantic error  
}
```

Error

```
error: lvalue required as left operand of assignment  
a + b = c; //semantic error
```

This article is contributed by **Krishna Bhatia**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](https://contribute.geeksforgeeks.org) or mail your article to [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org). See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



Practice Tags : C CPP

Article Tags : C++ C++ - Class 11 script java

An advertisement for a 'Free placement preparation course for IT Companies'. It features an illustration of a person sitting at a desk with a laptop and papers. The text 'Free placement preparation course for IT Companies.' is in the center. On the right, the 'Sudo Placement 2' logo is displayed with the tagline 'A computer science portal for geeks' and a 'Register Now!' button.

GeeksforGeeks  
A computer science portal for geeks

Free placement preparation course  
for IT Companies.

**Sudo  
Placement 2**  
Register Now!

Please write to us at [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org) to report any issue with the above content.

[Login to Improve this Article](#)

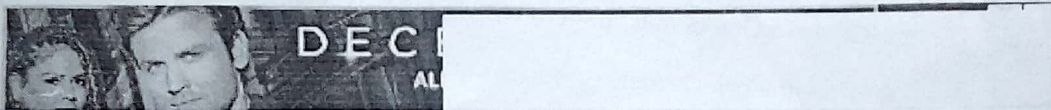
### Recommended Posts:

- Character arithmetic in C and C++
- Why is C considered faster than other languages ?
- C program to print characters without using format specifiers

<https://www.geeksforgeeks.org/errors-in-cc/>

```
// C program to illustrate
// syntax error
#include<stdio.h>
int main(void)
{
    // while() cannot contain "." as an argument.
    while(.)
    {
        printf("hello");
    }
    return 0;
}
```

Error:



```
error: expected expression before '.' token
while(.)
```

In the given example, the syntax of while loop is incorrect. This causes a syntax error.

2. **Run-time Errors** : Errors which occur during program execution(run-time) after successful compilation are called run-time errors. One of the most common run-time error is division by zero also known as Division error. These types of error are hard to find as the compiler doesn't point to the line at which the error occurs. For more understanding run the example given below.

```
// C program to illustrate
// run-time error
#include<stdio.h>
void main()
{
    int n = 9, div = 0;

    // wrong logic
    // number is divided by 0,
    // so this program abnormally terminates
    div = n/0;

    printf("resut = %d", div);
}
```

Error:

```
warning: division by zero [-Wdiv-by-zero]
div = n/0;
```

9/10/2018

In the given example, there is Division by zero error. This is an example of run-time error i.e errors occurring while running the program.

3. **Linker Errors:** These error occurs when after compilation we link the different object files with main's object using **Ctrl+F9** key(RUN). These are errors generated when the executable of the program cannot be generated. This may be due to wrong function prototyping, incorrect header files. One of the most common linker error is writing **Main()** instead of **main()**.

```
// C program to illustrate
// linker error
#include<stdio.h>

void Main() // Here Main() should be main()
{
    int a = 10;
    printf("%d", a);
}
```

Error:

```
(.text+0x20): undefined reference to 'main'
```

4. **Logical Errors :** On compilation and execution of a program, desired output is not obtained when certain input values are given. These types of errors which provide incorrect output but appears to be error free are called logical errors. These are one of the most common errors done by beginners of programming. These errors solely depend on the logical thinking of the programmer and are easy to detect if we follow the line of execution and determine why the program takes that path of execution.

```
// C program to illustrate
// logical error
int main()
{
    int i = 0;

    // logical error : a semicolon after loop
    for(i = 0; i < 3; i++);
    {
        printf("loop ");
        continue;
    }
    getchar();
    return 0;
}
```

No output

5. **Semantic errors :** This error occurs when the statements written in the program are not meaningful to the compiler.

```
// C program to illustrate
// semantic error
```

<https://www.geeksforgeeks.org/errors-in-c/>