



EDUCBA

(<https://www.educba.com/software-development/>)



(<https://www.educba.com/types-of-errors-in-c/>)



(<https://www.educba.com/preprocessor-in-c/>)



## Introduction to Programming Errors in C

Errors, in general, are referred to as an action or fault or problem which is incorrect or m... program behavior abnormal. In C programming language, the programming errors are bugs or faults that occur during runtime or compile time where the programs are not executed normally





(<https://www.educba.com/software-development/>)

obtain the desired outputs.

## Types of Programming Errors in C with Examples

In C, there are different types of errors that can occur in the programs which can make programs run abnormally. Errors are occurred by beginners such as run time, compile-time errors, warnings, etc can be corrected in different ways.

### Start Your Free Software Development Course

Web development, programming languages, Software testing & others

Given below are the types of programming errors that occur in C programs:

### 1. Syntax Errors

These are the errors that occur during compiling the programs. These errors are compile-time errors. As these syntax errors are thrown by the compilers during the program execution hence these syntax errors are called compilation errors. In general, these errors are made while writing programs or the syntax writing rules are not followed while writing the programs. Such errors can be easily corrected as they are known. These types of errors are usually made by beginners who are learning the programming language.

#### Example:

Suppose we want to declare any variable with particular data type then the correct method and syntax for declaring variable is as below:



System



(<https://www.educba.com/software-development/>)

Then the above declaration gives syntax error as to letter “l” for data type “int” is in the capital letter so.



🔗 Popular Course in this category



### C Programming Training (3 Courses, 5 Project)

3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access

★★★★★ 4.5 (8,635 ratings)

Course Price

**\$79** ~~\$399~~

[View Course](https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1)

(<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1>)

#### Related Courses

C++ Training (4 Courses, 5 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1>)

Java Training (40 Courses, 29 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1>)



Code:

```
#include <stdio.h>
```



(<https://www.educba.com/software-development/>)

```
}  
}
```

### Output:

Compilation failed due to following error(s).

```
main.c: In function 'main':  
main.c:4:5: error: unknown type name 'Int'  
    Int a = 10;  
    ^~~~~
```

In the above program, we saw the data type was written wrong so it gave compilation failed due to the error and due to the occurrence of this error the program could not be executed. The syntax error can be like not mentioning data type before the variable, or not terminating the printf statement with a semicolon (;), not ending the program with flower brackets ( {} ), etc.

## 2. Runtime Error

This error has occurred during runtime which means that occurs during program execution that is after compilation of the program. This error occurs mainly when the program is still running and it will not be able to perform some particular operation in the main which may lead to memory leakage.

### Example:

#### Code:

```
#include<stdio.h>  
  
intmain()  
{
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
}
```

**Output:**

```
main.c:6:18: warning: division by zero [-Wdiv-by-zero]
Floating point exception (core dumped)
```

The above program gives the divide by zero error during the program execution which can be handled by using exceptions.

### 3. Linker Error

This error has occurred when the executable file is not found or not created. This can be explained suppose if the main.c contains some function like func() whose executable file is func.c, this func() is defined in some other file such as sample.c and the object files generated are main.o and sample.o when the program is executing if the func() definition is not found in the sample.o then the linker will throw an error.

**Example:**

**Code:**

```
#include <stdio.h>
intMain()
{
```





(<https://www.educba.com/software-development/>)

## Output:

Compilation failed due to following error(s).

```
/usr/bin/x86_64-linux-gnu-ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): relocation 10 has invalid symbol index 13
/usr/bin/x86_64-linux-gnu-ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): relocation 11 has invalid symbol index 13
/usr/bin/x86_64-linux-gnu-ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): relocation 12 has invalid symbol index 13
/usr/bin/x86_64-linux-gnu-ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): relocation 13 has invalid symbol index 13
/usr/bin/x86_64-linux-gnu-ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): relocation 14 has invalid symbol index 13
/usr/bin/x86_64-linux-gnu-ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): relocation 15 has invalid symbol index 13
/usr/bin/x86_64-linux-gnu-ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): relocation 16 has invalid symbol index 13
/usr/bin/x86_64-linux-gnu-ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): relocation 17 has invalid symbol index 13
/usr/bin/x86_64-linux-gnu-ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): relocation 18 has invalid symbol index 13
/usr/bin/x86_64-linux-gnu-ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_info): relocation 19 has invalid symbol index 21
/usr/bin/x86_64-linux-gnu-ld: /usr/lib/debug/usr/lib/x86_64-linux-gnu/crt1.o(.debug_line): relocation 0 has invalid symbol index 2
/usr/lib/gcc/x86_64-linux-gnu/6/../../../../x86_64-linux-gnu/crt1.o: In function `_start':
(.text+0x20): undefined reference to `main'
collect2: error: ld returned 1 exit status
```

In the above program, the error occurred because of writing “Main” instead of “main” which is the most common linker error.

## 4. Semantic Error

These errors have occurred when the program statements are not correctly written which will make the compiler difficult to understand such as the use of the uninitialized variable, type compatibility, errors in writing expressions, etc.

### Example:

### Code:

```
#include <stdio.h>

intmain()

{
```





(<https://www.educba.com/software-development/>)

```
x+y=res;  
return 0;  
}
```

Output:

Compilation failed due to following error(s).

```
main.c: In function 'main':  
main.c:8:8: error: lvalue required as left operand of assignment  
    x+y=res;  
        ^
```

In the above program, the variable “res” cannot be used twice as the left operand.

## 5. Logical Error

This is an error that gives incorrect output due to illogical program execution but they seem to be error-free hence they are called logical errors.

Example:

Code:

```
#include <stdio.h>  
int main()  
{
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
sum=sum+a;
a++;
}
printf("The sum of the numbers is %d", sum);
return 0;
}
```

#### Output:

```
The sum of the numbers is 1
```

The above program is to print the output as the sum of numbers less than 20, but as we have specified semicolon after the for loop statement so it gives the wrong output where statements inside for loop are not executed. Hence the logical error occurs by printing the sum of 20 numbers as 1.

## Conclusion

In this article, we have discussed the programming errors that are occurred in C programming languages. To conclude with this article is that there are different types of errors that occur, which can be occurred due to the writing the programs incorrectly, or by writing the wrong syntax, or by writing wrong expressions in programs, or by not following the rules of writing

programs such as case sensitive is not followed, etc. In this article, we have discussed the errors such as runtime errors, compile-time errors, syntax errors, semantic errors, and logical errors.







[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

a look at the following articles to learn more –

1. [Arrays in C Programming \(https://www.educba.com/arrays-in-c-programming/\)](https://www.educba.com/arrays-in-c-programming/)
2. [Leap Year Program in C \(https://www.educba.com/leap-year-program-in-c/\)](https://www.educba.com/leap-year-program-in-c/)
3. [Constants in C \(https://www.educba.com/constants-in-c/\)](https://www.educba.com/constants-in-c/)
4. [Types of Errors in C \(https://www.educba.com/types-of-errors-in-c/\)](https://www.educba.com/types-of-errors-in-c/)

## ALL IN ONE SOFTWARE DEVELOPMENT BUNDLE (600+ COURSES, 50+ PROJECTS)

- ☒ 600+ Online Courses
- ☒ 50+ projects
- ☒ 3000+ Hours
- ☒ Verifiable Certificates
- ☒ Lifetime Access

**Learn More**

<https://www.educba.com/software-development/courses/software-development-course/?btnz=edu-blg-inline-banner3>





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Sign Up (<https://www.educba.com/software-development/signup/?source=footer>)

Corporate Training (<https://www.educba.com/corporate/?source=footer>)

Certificate from Top Institutions (<https://www.educba.com/educbalive/?source=footer>)

Contact Us (<https://www.educba.com/contact-us/?source=footer>)

Verifiable Certificate (<https://www.educba.com/software-development/verifiable-certificate/?source=footer>)

Reviews (<https://www.educba.com/software-development/reviews/?source=footer>)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

## Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)

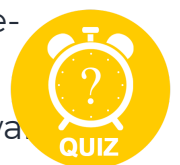
## Resources

Free Courses (<https://www.educba.com/software-development/free-courses/?source=footer>)

Java Tutorials (<https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer>)

Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)

All Tutorials (<https://www.educba.com/software-development/software-development-tutorials/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

[development-course/?source=footer\)](https://www.educba.com/software-development/courses/python-certification-course/?source=footer)

Become a Python Developer (<https://www.educba.com/software-development/courses/python-certification-course/?source=footer>)

Java Course (<https://www.educba.com/software-development/courses/java-course/?source=footer>)

Become a Selenium Automation Tester (<https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer>)

Become an IoT Developer (<https://www.educba.com/software-development/courses/iot-course/?source=footer>)

ASP.NET Course (<https://www.educba.com/software-development/courses/asp-net-course/?source=footer>)

VB.NET Course (<https://www.educba.com/software-development/courses/vb-net-course/?source=footer>)

PHP Course (<https://www.educba.com/software-development/courses/php-course/?source=footer>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

