# **روش پیاده‌سازی**

زبان برنامه‌نویسی C++ یک زبان برنامه‌نویسی با استفاده از یک کامپایلر کد را به کد ماشینی ترجمه می‌کند. این فرآیند شامل مراحل مختلفی می‌شود:

۱. نوشتن کد: شروع به نوشتن کد C++ می‌کنید. این کد می‌تواند در یک فایل منبع با پسوند cpp باشد.  
۲. کامپایل: سپس از یک کامپایلر C++ برای ترجمه کد به کد ماشینی استفاده می‌شود. این مرحله شامل چک کردن اشکال نحوی و ساختاری کد نیز می‌شود.

۳. تولید فایل اجرایی: در این مرحله، کامپایلر خروجی را به یک فایل اجرایی تبدیل می‌کند که می‌تواند در سیستم‌عامل اجرا شود.

C++ به عنوان یک زبان کامپایل شونده در عملکرد بهینه‌تر و سرعت اجرای بالاتری نسبت به زبان‌های تفسیری دارد، اما در عین حال، پیاده‌سازی و استفاده از زبان‌های تفسیری در مواردی مانند توسعه سریع و تست کد می‌تواند مزایای خود را داشته باشد.

# **خطاهای سی پلاس پلاس**

در این زبان خطاها به دو دسته اصلی تقسیم می‌شوند: خطاهای کامپایل و خطاهای اجرا

1. خطاهای کامپایل: این خطاها در مرحله کامپایل کد ایجاد می‌شوند و جلوی ترجمه کد به ماشینی را می‌گیرند.

مثال

* خطای نحوی (Syntax Error): مشکلات در ساختار و نحو کد که توسط کامپایلر تشخیص داده می‌شوند.

   int main() {  
       int x;  
       y = 10;  // خطا: متغیر y تعریف نشده است  
       return 0;  
   }

* خطای نوع داده (Type Error): استفاده از نوع داده‌ای نامعتبر یا نادرست.

     int main() {  
         int x = "Hello";  // خطا: نوع داده نامعتبر  
         return 0;  
     }

* خطاهای نگارشی(Typographical Errors) :

این خطاهای نگارشی معمولاً در مرحله کامپایل با پیغام‌های واضحی همراه می‌شوند که به برنامه‌نویس کمک می‌کنند تا مشکلات را سریعاً تصحیح کند.

* سمی‌کالن (Semicolon) نگذاشتن:

   int main() {  
       std::cout << "Hello World"  // خطا: سمی‌کالن ناقص  
       return 0;  
   }

* کم و زیاد بودن پرانتز:

   int main(  
       std::cout << "Hello World";  // خطا: پرانتز ناقص  
   }

* کم و زیاد بودن آکولاد:

   int main() {  
       std::cout << "Hello World";  // خطا: آکولاد ناقص

   int main()   
       std::cout << "Hello World";  // خطا: آکولاد ناقص

* تعاریف تکراری (Duplicate Definitions) :تعریف یک متغیر یا تابع در یک فایل بیش از یک بار.

   int x;  // تعریف اولیه  
   int x;  // خطا: تعریف تکراری

* استفاده نادرست از اشاره‌گرها: (Pointer Errors)

   int\* ptr;  
   \*ptr = 10;  // خطا: اشاره‌گر به مکانی اشاره نمی‌کند

* استفاده از نام تابع به جای نام متغیر و بالعکس:

   int main() {  
       int cout = 5;  
       std::cout << cout;  // خطا: استفاده نادرست از نام cout  
       return 0;  
   }

* تداخل نام‌ها: (Namespace Conflict) تداخل نام‌ها در استفاده از فضای نام‌ها.

   namespace A {  
       int x;  
   }  
  
   namespace B {  
       int x;  
   }  
  
   int main() {  
       using namespace A;  
       using namespace B;  
       std::cout << x;  // خطا: تداخل نام‌ها  
       return 0;  
   }

* خطای تداخل نوع داده: (Type Clash)

   int x = "Hello";  // خطا: تداخل نوع داده

* خطاهای حافظه (Memory Errors): از جمله دسترسی به حافظه غیرمجاز یا حافظه ناتوان.

   int\* ptr = nullptr;  
   \*ptr = 10;  // خطا: دسترسی به حافظه غیرمجاز

* تعداد پارامترهای نادرست (Mismatched Number of Parameters): تعداد پارامترهای ارسال شده به یک تابع با تعداد پارامترهای تعریف شده مطابقت ندارد.

   void printMessage(std::string msg) {  
       // ...  
   }  
  
   // خطا: تعداد پارامترها مطابقت ندارد  
   printMessage("Hello", 10);

* مسائل نام‌گذاری (Naming Issues): استفاده از نام‌گذاری نادرست برای متغیرها یا توابع.

   int main() {  
       int 2x = 5;  // خطا: نام‌گذاری نادرست  
       return 0;  
   }

2. خطاهای اجرا: این خطاها در زمان اجرا برنامه ایجاد می‌شوند و ممکن است به علت ورود داده‌های نامعتبر یا اشکال در الگوریتم‌ها رخ دهند.  
 مثال

دسترسی به اندیس خارج از محدوده: (Out of Range) در زمان اجرا، دسترسی به یک اندیس خارج از محدوده یک آرایه

   #include <iostream>   int main() {  
       int arr[3] = {1, 2, 3};  
       std::cout << arr[5];  // خطا: دسترسی به اندیس خارج از محدوده  
       return 0;  
   }

* تقسیم بر صفر (Division by Zero): در زمان اجرا، تلاش برای تقسیم عددی بر صفر.

     #include <iostream>

     int main() {  
         int a = 10, b = 0;  
         std::cout << a / b;  // خطا: تقسیم بر صفر  
         return 0;  
     }

در هر دو حالت، خروجی کامپایلر یا اطلاعات اجرا می‌تواند شامل اطلاعات دقیق‌تری درباره خطاها باشد که کمک می‌کند تا خطاها را بهبود دهید.