

اشکال زدایی در C++

در هر زبان برنامه نویسی نوشتن برنامه‌ای که بتواند در اولین اجرا بدون اشکال عمل کند کار بسیار مشکلی است، در واقع مواجه شدن با خطاهای برنامه همیشه قابل انتظار است و البته رفع خطاهای برنامه (debugging) لذت برنامه‌نویسی را دو چندان میکند. در اینجا با فرض اینکه محیط برنامه‌نویسی زبان C++ بوده و از کامپایلر Turbo یا Borland استفاده می‌شود با بررسی تعدادی از خطاهای رایج برنامه‌نویسی که در هنگام کار با این کامپایلرها بروز پیدا می‌کند روش رفع این خطاها را ارائه دهیم.

مفهوم Error و Warning:

Error: به اشکالاتی از برنامه اطلاق میشود که مانع از کامپایل بدون نقص برنامه و در نتیجه اجرای آن میشود، برای اجرای برنامه باید تمامی اشکالات آنرا برطرف کرد. این اشکالات شامل اشتباهات دستوری زبان (error Syntax)، تایپ نادرست دستورات، استفاده از ساختارهای نادرست برنامه نویسی و ... میشود.

Warning: مواردی است که از طرف کامپایلر به ما گوشزد میشود تا پیش از اجرای برنامه به آنها توجه کنیم. یک برنامه حتی با داشتن warning اجرا می‌شود ولی ممکن است نتیجه آن دور از انتظار ما باشد. این اشتباهات معنایی (errors Semantic) باید جدی گرفته شده و قبل از اجرای برنامه برطرف شوند.

نمونه هایی از خطاهای رایج

... function in 'X' symbol inedUnDef

۱- عدم تعریف فایل های سرآمد لازم برای اشیا پرکاربردی مثل cin و cout یا توابع کتابخانه ای.

۲- استفاده از یک شناسه (Identifier) که برای آن نوع (Type) تعریف نکرده ایم.

missing Statement ;

قرار ندادن سمی کالن در انتهای هر دستور (expression) منتهی به این خطا میشود.

missing statement Compound }

هر بلاک از دستورات که با "{" آغاز میشود باید با "}" خاتمه یابد، در تعریف توابع، دستورات شرطی و حلقه ها باید به این نکته توجه شود.

error taxsyn Declaration

۱- خطای رایج زمان برنامه نویسی که با تایپ اشتباه کلید واژه ها به وجود می آید.

۲- عدم توجه به ساختارهای دستوری C++ و رعایت نکردن قواعد گرامری این زبان

Unexpected }

۱- کاربرد اضافی }

۲- این error زمانی شکل میگیرد که بلاکی از دستورات بین "{" } قرار نگرفته باشد.

incorrectly terminated Declaration

۱- تابع main در برنامه وجود ندارد!

۲- در معرفی متغیرها/ توابع/ ثابت ها از شناسه ای نامعتبر استفاده شده است (برای مثال شناسه با عدد شروع شده یا یک کلیدواژه زبان است).

X file include open to Unable

احتمالاً X یک فایل سرآمد نیست یا اینکه در مسیری که فایل های سرآمد در آنجا ذخیره شده اند وجود ندارد یا نهایتاً در تایپ اسم آن اشتباهی صورت گرفته است.

پسوند ".h" برای فایل سرآمد X فراموش شده است.

directive preprocessor Unknown

به احتمال زیاد تعریف فایل سرآمد (دستور پیش پردازنده) با اشتباه تایپ کردن همراه بوده است ، مثلاً include درست تایپ نشده است.

missing Declaration

۱- شناسه به کار رفته برای متغیر/ تابع/ ثابت از قواعد زبان C++ پیروی نمیکند. برای مثال در آن از فاصله خالی (space) استفاده شده است.

constant character or string tedUntermina

۱- یک رشته باید بین یک جفت دابل کوتیشن قرار گیرد.

۲- یک ثابت کاراکتری بایستی بین یک جفت کوتیشن قرار گیرد.

long characters two or one be must constant Character

۱- احتمالاً یک رشته را بین یک جفت کوتیشن قرار داده اید، باید از دابل کوتیشن استفاده کنید.

۲- حداکثر طول یک ثابت کاراکتری یک یا دو حرف است.

pointer of use Illegal

۱- از عملگر "*" استفاده نادرست صورت گرفته است.

۲- در استفاده از اشاره گرها باید خیلی دقت نمود.

X character Illegal

به احتمال زیاد ساختار دستوری توضیح (tcommen) به کار رفته درست نیست.

operation structure Illegal

- ۱- برای عملیات ورودی خروجی از عملگرهای نادرست استفاده شده است.
- ۲- در یک عبارت از یک تابع void استفاده شده است (در حالی که توابعی از این نوع مقداری را برنمی گردانند)

syntax Expression

تایپ نادرست دستورات متداول (مثلاً <<<cout به جای <<cout استفاده شده است).

object const a modify Cannot

مقدار یک ثابت نمیتواند تغییر کند، پس تلاش بیهوده برای تغییر آن منتهی به یک error زمان کامپایل میشود.

required Lvalue

- ۱- از عملگر انتساب (=) به جای عملگر تساوی (==) استفاده شده است.
- ۲- قصد تغییر یک مقدار ثابت از نوع #define را داریم.

Y class for X operator generate not could Compiler

رعایت نکردن ساختار گرامری برای عملیات ورودی-خروجی (برای مثال هنگام کار با اشیایی مثل cin و cout)

Y to X convert Cannot

انتساب نادرست، برای مثال اگر به جای یک آدرس یک مقدار را به اشاره گری نسبت دهیم.

X for declaration Multiple

- ۱- اگر یک متغیر را چندین بار تعریف کنیم.
- ۲- اگر در یک تابع از یک نام یکسان برای دو یا چند متغیر استفاده کنیم.

indirection Invalid

انتساب غیر مستقیم یک مقدار به یک متغیر با استفاده از اشاره گری که به آن متغیر اشاره میکند "indirection" نام دارد. در اینجا این عمل به درستی صورت نگرفته است.

value a return cannot main

در اینجا تابع main (یا هر تابع دیگری) چون به صورت void معرفی شده است قادر به بازگرداندن مقداری به وسیله دستور return نیست.

X function undefined to Call

- ۱- نام تابع که در کتابخانه استاندارد (library standard) قرار دارد درست تایپ نشده است.
- ۲- فایل سرآمد لازم برای این تابع کتابخانه ای قید نشده است.

X tionfunc for defined been already has Body

- ۱- این خطا هنگامی بروز میکند که ما یک تابع را بیش از یک مرتبه تعریف کنیم.

X for match a find not Could

- در اینجا ما قصد داریم تابعی را احضار کنیم که قبلاً آنرا تعریف نکرده ایم.

X function to call in parameters few Too

- پارامترهای تابع مورد نظر بیشتر از تعداد آرگومان هایی است که در احضار تابع استفاده شده است.

X function to call in parameters Extra

- آرگومان هایی که برای احضار تابع استفاده شده از پارامترهای تابع بیشتر است.

X parameter following missing value Default

- در پیش الگو یا تعریف یک تابع از هر پارامتری که مقداری پیش فرض را آغاز کنیم پارامترهای بعدی نیز باید مقدار پیش فرض بگیرند که این امر در اینجا به درستی صورت نگرفته است.

here member class a intialize Cannot

- اگر بخواهیم به یک متغیر عضو کلاس مقدار اولیه دهیم (بدون در ارتباط بودن با یک شی) با خطای زمان کامپایل بالا روبرو میشویم.

here defined be maynot X Type

- ۱- تعریف کلاس یا ساختار به درستی صورت نگرفته است.
- ۲- سمیکالون (;) در انتهای ساختمان داده هایی مثل کلاس یا ساختار قرار میگیرد. در هردوی این حالت ها نوع جدیدی که ساختار یا کلاس تعریف میکند قابل استفاده نیست.

label missing statement Goto

- زمانی که از دستور goto استفاده شود بدون اینکه به برچسب (label) خاصی اشاره شود یا اینکه برچسبی وجود نداشته باشد.

تعدادی از Warning های رایج:

value a return should Function

تابع طوری تعریف شده که باید یک مقدار برگرداند (تابع غیر void) در حالی که چنین کاری انجام نشده است.

used never but value declared is X

X متغیری است که اعلان شده ولی در طول برنامه مورد استفاده قرار نگرفته است.

used never is that value assigned an is X

X متغیری است که تعریف شده (اعلان شده و مقدار گرفته) ولی در طول برنامه مورد استفاده قرار نگرفته است.

خطاهایی از این قبیل معمولاً در اثر فراموش کردن استفاده از متغیرها یا به کار گرفتن متغیرهای اضافه بر نیاز برنامه صورت میگیرد که حالت دوم زمانی رخ میدهد که برنامه بدون طرح قبلی و فی البداهه نوشته شود.

effect no has Code

استفاده از عملگرهای نامناسب، برای مثال ما در یک اشتباه بسیار رایج در دستورات مقایسه ای از عملگر "=" به جای عملگر "==" استفاده میکنیم.