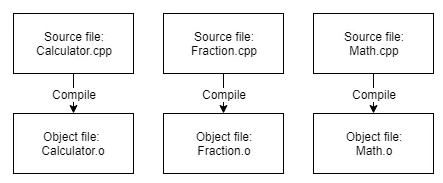
**Compiler and Linker**

برای کامپایل کردن یک برنامه ++C، از کامپایلر ++C استفاده می کنیم. کامپایلر ++C به طور متوالی از فایل های کد منبع (.cpp) در برنامه شما عبور می کند و دو کار مهم را انجام می دهد:

ابتدا کد شما را بررسی می کند تا مطمئن شود از قوانین زبان ++C پیروی می کند. اگر این کار را نکرد، کامپایلر به شما یک خطا (و شماره خط مربوطه) می دهد تا به شما کمک کند که چه چیزی نیاز به تعمیر دارد. فرآیند کامپایل نیز تا رفع خطا متوقف خواهد شد.

دوم، کد منبع ++C شما را به یک فایل زبان ماشین به نام object file ترجمه می کند. object file ها معمولاً name.o یا name.obj نامیده می‌شوند، جایی که نام آن ها همان نام فایل cpp. است که از آن تولید شده است.

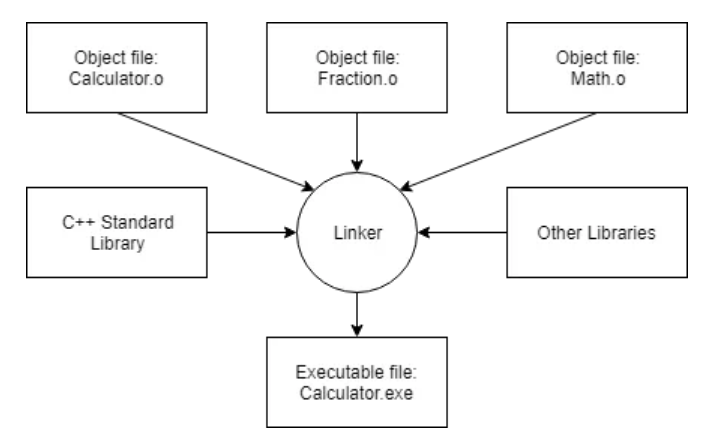
اگر برنامه شما دارای 3 فایل cpp. باشد، کامپایلر 3 فایل object تولید می کند:



**Linking object files and libraries**

پس از اینکه کامپایلر یک یا چند object file ایجاد کرد، برنامه دیگری به نام لینکر شروع به کار می کند. لینکر سه وظیفه دارد:

ابتدا، تمام فایل های شی تولید شده توسط کامپایلر را برمی دارد و آنها را در یک برنامه اجرایی واحد ترکیب می کند.

ثانیاً، علاوه بر اینکه می‌تواند object file ها را پیوند دهد، لینکر قادر است فایل‌های کتابخانه را نیز پیوند دهد. یک فایل کتابخانه ای مجموعه ای از کدهای از پیش کامپایل شده است که برای استفاده مجدد در برنامه های دیگر "بسته بندی" شده است.

با این حال، ++C همچنین با یک کتابخانه گسترده به نام کتابخانه استاندارد ++C ارائه می شود که عملکردهای اضافی را ارائه می دهد که می توانید از آنها در برنامه های خود استفاده کنید.یکی از پرکاربردترین بخش‌های کتابخانه استاندارد ++C، کتابخانه iostream است که دارای قابلیت چاپ متن روی مانیتور و دریافت ورودی صفحه‌کلید از کاربر را داراست.تقریباً هر برنامه ++C نوشته شده از کتابخانه استاندارد به شکلی استفاده می کند، بنابراین طبیعی است که کتابخانه استاندارد به برنامه های شما پیوند داده شود.اکثر لینکر ها به محض اینکه از قسمتی از کتابخانه های استاندارد استفاده می کنید به صورت خودکار لینک میشن به آن اکثر لینک‌کننده‌ها به‌محض اینکه از هر بخشی از کتابخانه استاندارد استفاده می‌کنید، به‌طور خودکار در کتابخانه استاندارد پیوند داده می‌شوند، بنابراین این چیزی نیست که نگران آن باشید.

سوم، لینکر مطمئن می شود که تمام وابستگی های بین فایل ها به درستی مقرر شده اند. به عنوان مثال، اگر چیزی را در یک فایل cpp. تعریف کنید، و سپس آن را در یک فایل .cpp دیگر استفاده کنید، لینکر این دو را به هم متصل می کند. اگر لینکر نتواند یک رفرنس را با تعریف آن به چیزی متصل کند، با یک خطای لینکر مواجه می شوید و فرآیند پیوند متوقف می شود.

هنگامی که لینکر تمام فایل‌ها و کتابخانه‌های شی را پیوند داد (با فرض اینکه همه چیز خوب پیش برود)، یک فایل اجرایی خواهید داشت که می‌توانید آن را اجرا کنید!

برای پروژه‌های پیچیده، برخی از محیط‌های توسعه از یک makefile استفاده می‌کنند، که فایلی است که نحوه ساخت یک برنامه را توضیح می‌دهد (مثلاً کدام فایل‌ها را کامپایل و پیوند دهید، یا به چه روش‌هایی پردازش کنید).