

| Time | Narration |
|-------|---|
| 00:01 | به برنامه آموزشی Scope of variables in C and C ++ خوش آمدید. |
| 00:08 | در این برنامه یاد می‌گیریم که: |
| 00:11 | محدوده متغیر (variable) چیست |
| 00:13 | Global variable (متغیرهای سراسری) چیست |
| 00:16 | Local variable چیست |
| 00:19 | و تعدادی مثال |
| 00:22 | و تعدادی از اشتباهات رایج و راه حل آن را توضیح می‌دهیم |
| 00:27 | برای ضبط این برنامه من از |
| 00:30 | Ubuntu operating system نسخه 11/04 و کامپایلر gcc و ++g نسخه 4/6/1 استفاده می‌کنم |
| 00:41 | با مقدمه‌ای در مورد محدوده متغیرها آغاز می‌کنیم. |
| 00:47 | این محدوده‌ای از کد می‌باشد که در آن به متغیرها می‌توان دسترسی پیدا کرد. |
| 00:54 | بستگی به نوع آن و جای اعلام شده آن به دو دسته تقسیم می‌شود |
| 00:59 | Global variable (متغیرهای سراسری) و |
| 01:02 | Local variable (متغیرهای محلی) |
| 01:05 | حالا مثالی را می‌بینیم |
| 01:07 | من قبلاً این برنامه را در editor تایپ کرده‌ام. |
| 01:10 | آن را باز می‌کنم. |
| 01:14 | نام فایل ما scope.c می‌باشد. |
| 01:19 | حالا کد را توضیح می‌دهم. |
| 01:23 | این header file ما می‌باشد. |
| 01:26 | در اینجا ما دو متغیر سراسری a و b را اعلام کرده ایم. |
| 01:32 | و به آن‌ها مقادیر اولیه 5 و 2 را می‌دهیم. |
| 01:39 | متغیرهای سراسری (global) در دسترس تمامی عملگرهای برنامه می‌باشد. |
| 01:44 | این‌ها بیرون از هر عملگری بالای عملگر main() اعلام می‌شوند. |
| 01:51 | این‌ها global scope دارند. |
| 01:53 | در اینجا ما function add بدون arguments را نشان داده ایم |
| 01:59 | در اینجا sum متغیر local می‌باشد که در عملگر add اعلام شده است. |
| 02:07 | متغیر local فقط برای عملگری که در آن اعلام شده است در دسترس می‌باشد |
| 02:13 | این متغیرها در داخل قالب اعلام شده اند |
| 02:16 | و آن‌ها local scope هستند |
| 02:19 | بنابراین جمع a و b در متغیر sum ذخیره خواهد شد. و اینجا جمع را چاپ می‌کنیم. |

| | |
|-------|--|
| 02:29 | این عملگر () main ما می باشد. |
| 02:33 | عملگر add فراخوانی و سپس اجرا می شود. |
| 02:38 | و این دستور return می باشد. |
| 02:40 | حال Save را کلیک کنید. |
| 02:43 | حال برنامه را اجرا می کنیم. |
| 02:45 | پنجره ترمینال را با فشار دادن همزمان Ctrl, Alt و کلید T در صفحه کلید باز کنید. |
| 02:55 | برای کامپایل gcc space scope.c space hyphen o space sco را تایپ کرده و Enter را فشار دهید. |
| 03:05 | برای اجرا ./sco را تایپ و Enter را فشار دهید. |
| 03:10 | خروجی به صورت |
| 03:13 | Sum of a and b is 7 نمایش داده می شود. |
| 03:16 | حال همین برنامه را در C++ اجرا می کنیم. |
| 03:20 | به برنامه مان برمیگردیم ابتدا کلیدهای Shift, Ctrl و S را همزمان فشار دهید. |
| 03:31 | حال فایل را با امتداد cpp ذخیره کنید. Save را کلیک کنید |
| 03:41 | فایل (کتابخانه) Header را به iostream تغییر می دهیم. |
| 03:47 | حال دستور Using را شامل کنید. Save را کلیک کنید. |
| 03:58 | اعلام متغیر های Global و Local در C++ به همین صورت می باشد. |
| 04:03 | پس نیازی به هیچ تغییری نیست. |
| 04:07 | حال دستور cout را جایگزین دستور printf کنید. |
| 04:13 | و format specifier و \n را حذف کنید |
| 04:17 | کاما را حذف کنید |
| 04:19 | دو کروشه زاویه ای باز را تایپ کنید. |
| 04:22 | کروشه بسته را حذف کنید. دوباره دو کروشه زاویه ای باز را تایپ کنید. |
| 04:26 | بین نقل قول دوگانه backslash n را تایپ کنید. حال Save را کلیک کنید. |
| 04:35 | حال برنامه را اجرا کنید. |
| 04:39 | به ترمینال خود باز می گردیم. |
| 04:42 | برای کامپایل gcc space scope dot cpp space -o Space sco1 را تایپ کنید |
| 04:52 | در اینجا ما sco1 را داریم چون نمیخواهیم که پارامتر خروجی sco را برای فایل scope.c بازنویسی کنیم. |
| 05:04 | Enter را فشار دهید. |
| 05:07 | برای اجرا ./sco1 را تایپ کنید و Enter را فشار دهید. |
| 05:14 | خروجی به صورت Sum of a and b is 7. نمایش داده می شود |
| 05:19 | میبینیم که این مثل کدهای C می باشد. |
| 05:27 | حال تعدادی از اشتباهات رایج را بررسی می کنیم. |

| | |
|-------|---|
| 05:31 | به برنامه خود باز می گردیم. فرض کنید در اینجا دوباره متغیر a را نشان می دهیم. |
| 05:41 | و a و semicolon را تایپ کنید. |
| 05:45 | Save را کلیک کنید. ما متغیر a را بالای عملگر main() و بعد از عملگر add نشان داده ایم. |
| 05:55 | بینیم چه می شود. |
| 05:57 | به ترمینال خود باز می گردیم. |
| 06:01 | حالا مثل قبل کامپایل کنید. |
| 06:05 | اشباهاتی می بینیم redefinition of 'int a', 'int a' previously defined here. به برنامه خود باز می گردیم. |
| 06:18 | و a متغیر global می باشد. |
| 06:20 | و global scope دارد. |
| 06:22 | ما متغیر را نمی توانیم دو بار اعلام کنیم چون از قبل به صورت سراسری (global) اعلام شده است. |
| 06:27 | ما variable a را به صورت متغیر local می توانیم نشان دهیم. |
| 06:34 | و اشتباه را تصحیح می کنیم. |
| 06:36 | این را حذف کنید. |
| 06:39 | Save را کلیک کنید. |
| 06:41 | دوباره اجرا می کنیم. و به ترمینال خود باز می گردیم. |
| 06:45 | مثل قبل کامپایل و اجرا کنید. |
| 06:49 | و انجام می شود. |
| 06:52 | این ما را به پایان این برنامه می آورد |
| 06:56 | به طور خلاصه این موارد را یاد گرفتیم |
| 06:58 | |
| 07:00 | محدوده متغیر |
| 07:02 | متغیرهای سراسری (global) مثل <code>& int a=5</code> |
| 07:07 | متغیرهای local مثل <code>int sum</code> |
| 07:12 | و حالا ارائه: |
| 07:14 | یک برنامه برای تفریق دو عدد بنویسید. |
| 07:19 | ویدیو را در لینک زیر مشاهده کنید. |
| 07:22 | این خلاصه پروژه Spoken Tutorial میباشد |
| 07:25 | اگر پهنای باند خوبی ندارید، ابتدا دانلود و سپس مشاهده کنید |
| 07:30 | تیم پروژه "Spoken Tutorial" |
| 07:32 | کارگاه آموزشی استفاده از "Spoken Tutorial" ارائه میدهد |
| 07:35 | و به کسانی که آزمون آنلاین را قبول شوند گواهینامه میدهد |
| 07:40 | برای جزئیات بیشتر لطفاً به " contact@spoken-tutorial.org " ایمیل بفرستید |

| | |
|-------|--|
| 07:47 | Spoken tutorial بخشی از پروژه Talk to a Teacher می باشد. |
| 07:52 | که تحت پشتیبانی National Mission on Education از طریق ICT ، توسط MHRD دولت هند می باشد |
| 08:00 | اطلاعات بیشتر درلینک نشان داده شده موجود می باشد |
| 08:04 | ترجمه و صداگذاری توسط شبیم اقبال خان.. |
| 08:08 | با تشکر از شما. |