Logical Operators

Time	Narration
00:02	.خوش آمدید Logical operators in C and C++ به برنامه آموزشی
00:08	expression1 && expression2 مثلاً logical AND یاد می گیریم. مثل logical در این برنامه در مورد اپراتورهای
00:16	expression1 OR expression2 مثل Logical OR
00:21	not (Expression1) مثل Logical NOT
00:25	.و این را با کمک مثال انجام می دهیم
00:28	نسخه 4/6/1 g++ و gcc نسخه 11/10 وکامپایلر Ubuntu Operating System برای ضبط این برنامه من از
00:33	.استفاده کرده ام ubuntu در
00:39	.آغاز می کنیم logical operators با مقدمه‌ای در مورد
00:43	.هر مقدار غیر از 0 می باشد true و C++, و C در
00:48	true غیر از صفر یعنی
00:50	false و صفر یعنی
00:53	.می دهند false و 0 را برای true را استفاده می‌کنند 1 را برای Logical operators عباراتی که
00:58	.را با کمک مثالی توضیح می دهم Logical operators حالا
01:03	.می باشد C در Logical operators در اینجا یک برنامه برای
01:08	main در قسمت
01:10	.را بعنوان عدد صحیح اعلام می کند c و a,b این دستور, متغیرهای
01:16	.را وارد کند c و a, b از کاربر می‌خواهد که مقدار printf دستور
01:21	.را از کاربر می گیرد c و a, b ورودی های متغیرهای scanf دستور
01:28	.را برای پیدا کردن بیشترین مقایسه می کنیم cو b و a در اینجا ما مقدار
01:33	.استفاده می کنیم logical AND برای مقایسه همزمان از اپراتور
01:38	.را بدهد true باید درست باشد تا logical AND در اینجا تمام حالتها برای
01:43	.در صورت حالت اشتباه, عبارت ارزیابی نمی شود
01:49	.درست باشد ارزیابی می شود (a>b) اگر (a>c) پس عبارت

01:56	.باشد دیگر عبارت ارزیابی نمی شود b کمتر از a اگر
02:02	.این دستور در صورتی که حالت قبلی درست باشد ارزیابی می شود
02:07	.ارزیابی میشود (b>c) سپس
02:10	.روی صفحه نمایش داده می شود b is greatest اگر حالت درست باشد
02:16	.روی صفحه نمایش داده می شود c is greatest در غیر این صورت
02:21	.می رویم logical OR حالا به اپراتور
02:24	.مقدار درست را بدهد logical OR اینجا یکی از حالتها باید درست باشد تا
02:30	.در صورت حالت درست , عبارت بعداً ارزیابی نمی شود
02:35	.پس دو عبارت دیگر ارزیابی نخواهند شد a==zero اگر
02:43	.اجرا می شود printf 0 باشد دستور (صفر) cیا a, b اگر یکی از
02:49	و آکولاد بسته return 0 به پایان برنامه می‌آییم
02:54	.حالا برنامه را ذخیره کنید
02:57	.کنید save (ذخیره) .c آن را با امتداد
03:00	.ذخیره کرده ام logical.c من فایل را با نام
03:03	.باز کنید T و Ctrl,Alt پنجره ترمینال را با فشار دادن همزمان کلیدهای
03:08	.را فشار دهید Enter .را تایپ کنید gcc space logical dot c space minus o space log برای کامپایل کد
03:23	.را تایپ کنید ./log برای اجرا
03:27	.را فشار دهید Enter
03:29	.من مقادیر 0, 34, 567 را وارد می کنم
03:39	.خروجی نمایش داده می شود
03:42	C is greatest
03:45	.صفر می باشد c و a, b حاصل ضرب
03:50	.شما این برنامه را با ورودی های دیگر اجرا کنید
03:55	.می نویسیم C++ حالا همین برنامه را در
03:59	.من از قبل برنامه را نوشته ام
04:03	.می باشند C++ در اینجا کدها در
04:06	.می نویسیم. تغییراتی می دهیم C++ حالا همین برنامه را در

04:11	.می باشد header تغییر در فایل
04:14	.را استفاده کرده ام using دستور
04:18	.همچنین در دستورهای ورودی و خروجی تغییراتی می باشد
04:21	.عمل می کنند C اپراتورها با همان روش در
04:25	.را کلیک کنید Save
04:27	.ذخیره کنید .cpp فایل را با امتداد
04:31	.باز کنید T و Ctrl,Alt ترمینال را با فشار دادن همزمان کلیدهای
04:36	.را فشار دهید Enter.را تایپ کنید g++ logical.cpp space minus o space log1 برای کامپایل
04:49	.را تایپ کنید ./log1 برای اجرا
04:53	.را فشار دهید Enter
04:56	.من مقادیر 0, 34, 567 را وارد می کنم
05:02	.می باشد C می‌بینیم که خروجی مثل همان در برنامه
05:05	.برنامه را با ورودی های دیگر اجرا کنید
05:10	.حالا یک اشتباه متداول را بررسی می کنیم
05:12	.باز می گردیم editor به
05:16	.فرض کنید در اینجا براکتها را فراموش کرده ام
05:20	.این و این را حذف کنید
05:26	.حالا ببینیم چه می شود. برنامه را ذخیره کنید
05:30	.به ترمینال بازگردید
05:32	.مثل قبل کامپایل و اجرا کنید
05:38	.این اشتباه را می بینیم
05:41	Expected identifier before ‘(‘ token.
05:45	.زیرا در اینجا دو عبارت متفاوت داریم
05:48	.ارزیابی کنیم AND ما باید آن‌ها را بعنوان یک عبارت با استفاده از اپراتور
05:53	.حالا به برنامه برمی‌گردیم و اشتباه را تصحیح می کنیم
05:57	.براکت را اینجا و اینجا وارد کنید
06:04	.را کلیک کنید Save

06:06	.به ترمینال باز گردید
06:09	.مثل قبل کامپایل و اجرا کنید
06:14	.و عمل می کند
06:22	.حالا برنامه را خلاصه می کنیم
06:24	((a>c) && (a>b)) مثل logical AND.در این برنامه این موارد را یاد گرفتیم
06:32	(a==0 || b==0 || c==0) مثل Logical OR
06:39	:ارائه
06:41	.یک برنامه که دو عدد را بعنوان ورودی از کاربر می‌گیرد بنویسید
06:44	.را استفاده کنید NOT بررسی کنید که آیا دو عدد مساوی هستند. اپراتور (a != b) :راهنمایی
06:54	.ویدیو را در لینک زیر مشاهده کنید
06:57	می باشد spoken tutorial این خلاصه پروژه
06:59	اگر پهنای باندخوبی ندارید، ابتدا دانلود وسپس مشاهده کنید
07:03	ارائه میدهد spoken tutorial کارگاه آموزشی استفاده از Spoken Tutorial تیم پروژه
07:07	و به کسانی که آزمون آنلین را قبول شوند گواهینامه میدهد
07:11	ایمیل بفرستید ''contact@spoken-tutorial.org''برای جزئیات بیشتر لطفا به
07:18	.می باشد Talk To a Teacher بخشی از پروژه Spoken tutorial
07:21	که تحت پشتیبانی National Mission on Education .دولت هند می باشد MHRD توسط ,ICT از طریق
07:27	موجود می باشد spoken hyphen tutorial dot org slash NMEICT hyphen intro اطلاعات بیشتر
07:30	
07:37	ترجمه و صداگذاری توسط شبنم اقبال خان...با تشکر از شما