

Narration	Time
C اور ++C میں logical operators پر اسپوکن ٹیوٹورل میں خوش آمدید۔	00:02
اس ٹیوٹوریل میں ہم logical آپریٹر، جیسے logical AND مثلاً expression1 && expression2 کے بارے میں سیکھیں گے۔	00:08
logical OR مثلاً expression1 or expression2	00:16
logical NOT مثلاً not (Expression1)	00:21
ہم مثالوں کی مدد سے یہ کریں گے۔	00:25
اس ٹیوٹوریل کو ریکارڈ کرنے کے لئے میں آپریٹنگ سسٹم کے طور پر اوپنٹو 11.10 استعمال کر رہا ہوں،	00:28
اوپنٹو پر "gcc" اور "g + +" کپائلر ورژن 4.6.1	00:33
logical آپریٹر کے تعارف کے ساتھ ہم شروع کرتے ہیں۔	00:39
C اور ++C میں، "true"، 0 کے علاوہ کوئی بھی دوسرا ویلیو ہے۔	00:43
non zero کا مطلب True ہے۔	00:48
zero کا مطلب False ہے۔	00:50
logical آپریٹر کا استعمال "true" Expressions کے لئے 1 اور "false" کے لئے 0 رٹرن کرتا ہے۔	00:53
اب میں ایک مثال کی مدد سے logical آپریٹر سمجھاؤں گا۔	00:58
یہاں "C" میں logical آپریٹر کے لئے ایک پروگرام ہے۔	01:03
"main" بلاک کے اندر،	01:08
یہ اسٹیٹمنٹ "a"، "b" اور "c" ویریبلز کو انچر کے طور پر ظاہر کرتا ہے۔	01:10
"printf" اسٹیٹمنٹ یوزر کو "a"، "b" اور "c" کی ویلیوز کو درج کرنے کے لئے اشارہ کرتا ہے۔	01:16
"scanf" اسٹیٹمنٹ یوزر سے ویریبل "a"، "b" اور "c" کے لئے ان پٹ لیتا ہے۔	01:21
یہاں، سب سے بڑی ویلیو تلاش کرنے کے لئے ہم "a" کی ویلیو کا مقابلہ "b" اور "c" سے کر رہے ہیں۔	01:28
ایک ساتھ موازنہ کرنے کے لئے ہم "logical AND" آپریٹر کا استعمال کرتے ہیں۔	01:33
یہاں، True ویلیو رٹرن کرنے کیلئے، "logical AND" کی تمام کنڈیشنس کو True ہونا چاہئے۔	01:38
False کنڈیشن کا سامنا کرنے پر expression مزید آگے نہیں بڑھتی۔	01:43

لہذا "a > c" expression کی اویلیوشن یعنی جانچ تب ہی ہوگی اگر "a > b" True ہے۔	01:49
اگر "a"، "b" سے کم ہے تو expression کی آگے کی جانچ نہیں ہوگی	01:56
اگر گزشتہ کنڈیشن True ہے تو اس اسٹیٹمنٹ کو اویلیوٹ کیا جاتا ہے۔	02:02
آگے (b > c) کی جانچ ہوتی ہے۔	02:07
اگر کنڈیشن true ہے، تو سکرین پر b is greatest ظاہر ہوتا ہے۔	02:10
دوسری صورت میں سکرین پر c is greatest ظاہر ہوتا ہے۔	02:16
اب ہم logical OR آپریٹر پر آتے ہیں۔	02:21
یہاں True ویلیورٹن کے لئے logical OR کے لئے کوئی ایک کنڈیشن True ہوگی۔	02:24
True کنڈیشن کا سامنا کرنے پر expression کی آگے کی جانچ نہیں ہوگی۔	02:30
لہذا، اگر a == zero ہے، تو باقی دو expressions کی جانچ نہیں ہوگی۔	02:35
اگر a، b یا c زیرو ہے تو یہ "printf" اسٹیٹمنٹ ایکز کیوٹ ہوتا ہے۔	02:43
پروگرام کے آخر میں آتے ہیں۔ return 0 اور کلوز کر لی بریکٹ۔	02:49
اب پروگرام کو سیو کریں۔	02:54
.c extension کے ساتھ فائل سیو کریں۔	02:57
میں نے اپنی فائل logical.c نام سے سیو ہے۔	03:00
Ctrl، Alt اور T کیز ایک ساتھ دبا کر ٹرمینل کھولیں۔	03:03
کوڈ کمپائل کرنے کے لئے ٹائپ کریں gcc space logical dot c space minus o space -log اینٹر دبا لیں۔	03:08
ایکز کیوٹ کرنے کے لئے ٹائپ کریں "./log" (dot slash log)	03:23
اینٹر دبا لیں۔	03:27
میں یہ ویلیوز درج کروں گا۔ 0 34 567	03:29
آؤٹ پٹ ظاہر ہوتا ہے۔	03:39
c is greatest	03:42
The product of a, b and c is zero.	03:45

آپکو ان پٹ کے مختلف سیٹس کے ساتھ اس پروگرام کو ایکز کیوٹ کرنے کی کوشش کرنی چاہئے۔	03:50
اب وہی پروگرام ++C میں لکھتے ہیں۔	03:55
میں نے پہلے ہی پروگرام بنایا ہے اور میں آپ کو بتاتا ہوں۔	03:59
یہاں ++C میں کوڈ ہے۔	04:03
اب یہی پروگرام ++C میں بنانے کے لئے ہم کچھ تبدیلی کرتے ہیں۔	04:06
یہاں ہیڈرفائل میں تبدیلی ہے۔	04:11
using سٹیٹمنٹ استعمال کیا گیا ہے۔	04:14
اور یہاں کے آؤٹ پٹ اور ان پٹ اسٹیٹمنٹس میں بھی فرق ہے۔	04:18
آپریٹو ویسے ہی کام کرتے ہیں جیسے وہ C میں کرتے ہیں۔	04:21
"Save" پر کلک کریں۔	04:25
یقینی بنائیں، کہ فائل extension .cpp کے ساتھ سیو ہے۔	04:27
Alt، Ctrl اور T کیز ایک ساتھ دبا کر ٹرمینل کھولیں۔	04:31
پروگرام کپائل کرنے کے لئے، ٹائپ کریں g + + logical.cpp space minus o space log1۔ اینٹر دبائیں۔	04:36
ایگز کیوٹ کرنے کے لئے ٹائپ کریں ./log1۔	04:49
اینٹر دبائیں۔	04:53
میں یہ ویڈیو درج کروں گا۔ 0 34 567۔	04:56
ہم C پروگرام کی مانند آؤٹ پٹ دیکھتے ہیں۔	05:02
آپکو ان پٹ کے مختلف سیٹس کے ساتھ پروگرام کو ایکز کیوٹ کرنے کی کوشش کرنا چاہئے۔	05:05
اب ہم ایک ایرر دیکھتے ہیں جو آسکتا ہے۔	05:10
ایڈیٹر پر جائیں	05:12
مان لیں، یہاں ہم بریکٹس بھول گئے۔	05:16
اسے اور اسے ڈیلیٹ کریں۔	05:20
دیکھتے ہیں، کیا ہوتا ہے۔ پروگرام سیو کریں۔	05:26

05:30	ٹرمینل پرواپس جائیں۔
05:32	پہلے کی طرح کمپائل اور ایکزیکوٹ کریں۔
05:38	ہم ایک ایرر دیکھتے ہیں:
05:41	Expected identifier before '(' token.
05:45	یہ اس لئے کہ یہاں ہمارے پاس دو مختلف expressions ہیں۔
05:48	ہمیں AND آپریٹر کا استعمال کرتے ہوئے ان کی جانچ ایک expression کے طور پر کرنا ہے۔
05:53	اب اپنے پروگرام پرواپس جائیں اور ایرر درست کریں۔
05:57	یہاں بریکپوائنٹس درج کریں۔
06:04	Save کلک کریں۔
06:06	ٹرمینل پرواپس آئیں۔
06:09	پہلے کی طرح کمپائل اور ایکزیکوٹ کریں۔
06:14	اب یہ کام کر رہا ہے۔
06:22	اب اس ٹیوٹوریل کا خلاصہ بیان کرتے ہیں۔
06:24	اس ٹیوٹوریل میں ہم نے مندرجہ ذیل کے بارے میں سیکھا logical AND جیسے ((a > b) && (a > c))
06:32	logical OR، جیسے (a == 0 b == 0 c == 0)
06:39	ایک مشق
06:41	ایک پروگرام لکھیں، جو پوزر سے ان پٹ کے طور پر دو نمبر لے۔
06:44	NOT آپریٹر کا استعمال کرتے ہوئے چیک کریں، کہ دو نمبر برابر ہیں یا نہیں۔ اشارہ ہے: (a != b)
06:54	مندرجہ ذیل لنک پر دستیاب ویڈیو دیکھیں۔
06:57	یہ اسپوکون ٹیوٹوریل پروجیکٹ کا خلاصہ اسٹیٹمنٹ کرتا ہے
06:59	اگر آپ کے پاس اچھی بینڈ ویڈتھ نہیں ہے تو آپ اسے ڈاؤن لوڈ کر کے بھی دیکھ سکتے ہیں۔
07:03	اسپوکون ٹیوٹوریل پروجیکٹ ٹیم، اسپوکون ٹیوٹوریل کا استعمال کرتے ہوئے ورکشاپ بھی چلاتی ہے۔

07:07	جو آن لائن ٹیسٹ پاس کرتے ہیں، انہیں سند بھی دیتے ہیں،
07:11	مزید معلومات کے لئے، contact@spoken-tutorial.org پر لکھیں۔
07:18	اسپوکن ٹیوٹوریل ٹاک ٹوائے ٹیچر پروجیکٹ کا حصہ ہے۔
07:21	یہ بھارت حکومت کے ایچ آر ڈی کے "آئی سی ٹی کے ذریعے قومی خواندگی مشن کی طرف سے حمایت شدہ ہے۔
07:27	اس مشن پر مزید معلومات اس لنک پر دستیاب ہیں۔
07:30	http://spoken-tutorial.org/NMEICT-Intro
07:37	اس اسکرپٹ کا ترجمہ اور صدابندی میں نے یعنی وجاحت احمد نے کی ہے، شامل ہونے کیلئے آپ کا شکریہ