Narration	Time
C اور + + C میں اریز(Arrays) پر سپوکن ٹیوٹوریل میں آپ کا استقبال ہے.	00.01
اس ٹیوٹوریل میں ہم سیکھیں گے،	00.07
ارے کیا ہے.	00.09
ارے کا اعلان یعنی ڈِکلیئر کرنا.	00.11
ارے کا انیشلا زیشن.	00.13
ارے پر کچھ مثالیں	00.16
ہم کچھ عام ایررس اور ان کے حل بھی دیکھیں گے.	00.18
اس ٹیوٹوریل کو ریکارڈ کرنے کے لئے، میں	00.22
اُبنٹو آپریٹنگ سسٹم ورژن 11.04،	00.25
gcc اور + +g کمپائلر ورژن 4.6.1 استعمال کر رہا ہوں.	00.30
ارے کے تعارف کے ساتھ شروع کرتے ہیں.	00.36
ارے ڈیٹا کا مجموعہ یا ایک ہی طرح کے ڈیٹا-ٹائپ کے ایلمینٹس ہیں	00.39
ارے انڈیکس 0 سے شروع ہوتا ہے.	00.44
پہلا ایلمینٹ انڈیکس 0 میں جمع ہے.	00.48
یہاں تین طرح کے اریز ہیں:	00.52
Single dimensional array (سنگل ڈامیشنل ارے)	00.55
Two dimensional array (ٹو ڈامیشنل ارے) اور	00.57
Multi-dimensional array (ملٹ -ڈامیشنل ارے)	00.59
اس ٹیوٹوریل میں ہم سنگل -ڈامیشنل ارے پر بات کریں گے.	01.01
دیکھتے ہیں، کہ سنگل ڈامیشنل ارے کیسے ڈِکلیئر کریں.	01.06
اس کے لئے سنٹیکس ہے:	01.09
data-type name of the array (ارے کا نام) اور size	01.11
مثلاً، یہاں ہم نے ایک انٹجر ارے سٹار ڈِکلیئر کیا ہے، جس میں 5 ایلمینٹس ہیں.	01.16

ارے انڈیکس، سٹار 0 سے سٹار 4 تک شروع ہو جائے گا.	01.24
ہم نے ارے کا ڈکلیئر کرنا دیکھا.	01.29
اب ہم ارے کا انیشلا زیشن دیکھیں گے.	01.32
اس کے لئے سنٹیکس ہے:	01.35
(ڈیٹا - ٹائپ، ارے کا نام، size is equal to elements ،name of the array، data-type سائز (سائیز) ایلمینٹس کے برابر ہے)	01.38
مثلاً، یہاں ہم نے سائز 3 کے ساتھ ایک انٹجر ارے سٹار کو ڈِکلیئر کیا ہے .ارے کے ایلمینٹس ہیں 1، 2 اور 3	01.44
یہاں انڈیکس سٹار 0 سے سٹار 2 تک شروع ہو جائے گا.	01.54
اب مثالوں پر آتے ہیں.	01.59
میں نے ایڈیٹر پر پہلے ہی پروگرام ٹائپ کیا ہے.	02.01
اب میں اسے کھولتا ہوں.	02.04
نوٹ کریں، کہ ہماری فائل کا نام array.c ہے.	02.06
اس پروگرام میں، ہم ارے میں جمع ایلمینٹس کے جوڑ کا حساب کریں گے.	02.10
اب، میں کوڈ سمجھاتا ہوں.	02.16
یہ ہماری ہیڈر فائل ہے.	02.18
یہ ہمارا مین فنکشن ہے.	02.20
یہاں ہم نے سائز 3 کے ساتھ ایک ارے سٹار (array star) کو ڈِکلیئر اور انشیلائز کیا ہے.	02.22
ارے کے ایلمینٹس ہیں 4، 5 اور 6	02.28
پھر ہم نے ایک انٹجر ویریبل sum، ڈِکلیئر کیا ہے.	02.33
یہاں ہم نے ارے کے ایلمینٹس جوڑے ہیں اور نتائج کو sum میں جمع کیا ہے.	02.36
نوٹ کریں، کہ 4 انڈیکس 0 میں جمع ہوگا، 5 انڈیکس 1 میں جمع ہوگا اور 6 انڈیکس 2 میں جمع ہوگا.	02.41
پھر ہم sum پرنٹ کریں گے.	02.50
اور یہ ہمارا return اسٹیٹمینٹ ہے.	02.52
اب، Save پر کلک کریں.	02.54

02.57	پروگرام ایکزِ کیوٹ کریں
02.59	براہ مہربانی اپنے کی بورڈ پر ایک ساتھ Ctrl ، Altاور T کیز دبا کر ٹرمینل ونڈو کھولیں.
03.09	کمپائل کرنے کے لئے ٹائپ کریں gcc space array dot c space hypen o space array اور اینٹر دبائیں.
03.19	ایکزِ کیوٹ کے لئے ٹائپ کریں .dot slash array، اینٹر دبائیں.
03.24	یہاں آوٹ پُٹ ظاہر ہوتا ہے،
03.26	The sum is 15.
03.28	اب کچھ عام ایررس دیکھتے ہیں، جو آ سکتے ہیں.
03.32	اپنے پروگرام پر آئیں.
03.34	مانیں کہ یہاں، لائن نمبر 4 پر ہم کرلی بریکیٹس بھول گئے ہیں.
03.39	Save پر کلک کریں . دیکھیں کیا ہوتا ہے.
03.42	ٹرمینل پر واپس آئیں.
03.44	پہلے کی طرح کمپائل کریں
03.47	ہم ایک ایرر دیکھتے ہیں.
03.49	Invalid initializer and Expected identifier or bracket before numeric constant.
03.56	یہ اس لئے کیونکہ اریز، کرلی بریکیٹس میں انشلائز کرنے ضروری ہیں.
04.01	اپنے پروگرام پر آئیں . ایرر فکس یعنی ٹھیک کریں.
04.04	یہاں لائن نمبر 4 پر کرلی بریکیٹس ٹائپ کریں.
04.09	اب، Save پر کلک کریں.
04.12	ایکزِ کیوٹ کریں، ٹرمینل پر واپس آئیں.
04.15	پہلے کی طرح کمپائل کریں۔ پہلے کی طرح ایکزِ کیوٹ کریں.
04.19	جی ہاں، یہ کام کر رہا ہے.
04.21	اب ہم اسی پروگرام کو + + C میں ایکزِ کیوٹ کریں گے.

اپنے پروگرام پر واپس آئیں.	04.25
میں یہاں کچھ تبدیلی کروں گا.	04.28
پہلے اپنے کی بورڈ پر ایک ساتھ Shift، Ctrl اور S کیز دبائیں.	04.30
اب extension dot cpp کے ساتھ فائل سیو کریں اور Save پر کلک کریں.	04.38
iostream کے طور پر ہیڈر فائل تبدیل کریں.	04.44
اب using اسٹیٹمینٹ شامل کریں	04.49
ارے کا ڈکلریشن اور انشیلازیشن C + + میں بھی اسی طرح ہی ہے.	04.55
اس لئے کچھ بھی تبدیل کرنے کی ضرورت نہیں ہے.	05.01
اب printf اسٹیٹمینٹ کو cout اسٹیٹمینٹ میں تبدیل کریں.	05.04
format specifier اور back slash n ڈلیٹ کریں، اب کوماڈلیٹ کریں اور دواوپننگ اینگل بریکیٹس ٹائپ کریں.	05.09
یہاں بریکٹ ڈلیٹ کریں۔ دوبارہ دواوپننگ اینگل بریکیٹس لکھیں اور ڈبل کوٹس میں back slash n ٹائپ کریں	05.17
Save پر کلک کریں.	05.26
ایکزِکیوٹ کریں۔ ٹرمینل پر واپس آئیں.	05.29
کمپائل کرنے کے لئے ٹائپ کریں، g+ + space array dot cpp space hypen o space array1.	05.32
یہاں ہمارے پاس ارے 1 ہے، کیونکہ array dot c فائل کے لئے ہم آوٹ پُٹ پیرامیٹر ارے کو اوور رائٹ کرنا نہیں چاہتے ہیں.	05.42
اب اینٹر دبائیں.	05.51
ایکزِکیوٹ کرنے کے لئے ٹائپ کریں، .dot slash array1 اینٹر دبائیں.	05.54
آوٹ پُٹ ظاہر ہوتا ہے، The sum is 15	05.59
ہم دیکھ سکتے ہیں، کہ یہ ہمارے C کوڈ کے جیسا ہی ہے.	06.02
اب، ہم دوسرے عام ایرر دیکھیں گے.	06.07
اپنے پروگرام پر واپس آئیں.	06.10

06.12	مانیں کہ یہاں، لائن نمبر 7 پر
06.14	میں ٹائپ کروں گا star(1)، star(2) اور star(3)؛
06.23	Save پر کلک کریں.
06.24	ایکزِکیوٹ کریں۔ اپنے ٹرمنل پر واپس آئیں.
06.28	پرومپٹ کلیئر کریں۔
06.30	پہلے کی طرح کمپائل کریں
06.33	پہلے کی طرح ایکزِکیوٹ کریں۔
06.36	ہمیں ایک غیر متوقع آوٹ پُٹ ملتا ہے.
06.39	یہ اس لئے کیونکہ ارے انڈیکس 0 سے شروع ہوتا ہے.
06.43	اپنے پروگرام پر واپس آئیں . ہم یہاں دیکھ سکتے ہیں، ارے انڈیکس 1 سے شروع ہوتا ہے.
06.49	اس لئے یہ ایرر دے رہا ہے . ایرر فکس کریں.
06.54	یہاں ٹائپ کریں0، 1اور 2، سیو پر کلک کریں.
07.02	ایکزِکیوٹ کریں۔اپنے ٹرمنل پر واپس آئیں.
07.05	پہلے کی طرح کمپائل کریں . پہلے کی طرح ایکزِکیوٹ کریں.
07.09	جی ہاں، یہ کام کر رہا ہے.
07.12	اب، ہم اپنی سلائڈ پر واپس جائیں گے.
07.14	مختصر میں...
07.16	اس ٹیوٹوریل میں ہم نے سیکھا،
07.19	اریز
07.20	Single dimensional array (سنگل ڈامیشنل اریز) ڈِکلیئر کرنا.
07.23	Single dimensional array (سنگل ڈامیشنل اریز) انشیلائز کرنا.
07.26	مثلا int star[3]={4, 5, 6}
07.31	ارے کے ایلیمینٹس کو جوڑنا، مثلاً sum is equal to star 0 plus star 1 plus star 2
07.40	مشق کے طور پر

07.41	ارے میں جمع ایلمینٹس کے تفریق کا حساب کرنے کے لئے ایک پروگرام لکھیں.
07.47	نیچے دکھائے گئے لنک پر دستیاب ویڈیو دیکھیں.
07.50	یہ سپوکن ٹیوٹوریل پروجیکٹ کو خلاصہ کرتا ہے.
07.53	اگر آپ کے پاس اچھی بینڈ وڈتھ نہیں ہے، تو آپ اسے ڈاؤن لوڈ کر کے بھی دیکھ سکتے ہیں.
07.57	سپوکن ٹیوٹوریل پروجیکٹ ٹیم
08.00	سپوکن ٹیوٹوریلس کا استعمال کرتے ہوئے ورکشاپ بھی چلاتی ہے.
08.03	آن لائن ٹیسٹ پاس کرنے والوں سند بھی دیتے ہیں.
08.06	مزید معلومات کے لئے contact@spoken-tutorial.org پر لکھیں.
08.13	سپوکن ٹیوٹوریل پروجیکٹ ٹاک -ٹو -اٹیچر پروجیکٹ کا حصہ ہے.
08.17	اسے قومی خواندگی مشن نے ICT، ایمپاررڈی حکومت ہند کے ذریعے حمایت کی ہے.
08.25	اس مشن پر مزید معلومات درج ذیل لنک پر دستیاب ہیں http://spoken-tutorial.org/NMEICT-Intro
08.30	اس اسکرپٹ کا ترجمہ اور صدا بندی میں نے یعنی وجاہت احمد نے کی ہے
08.33	ہمارے ساتھ شریک ہونے کیلیئے آپکا شکریہ