

Time	Narration
00:02	به spoken tutorial در First C++ program خوش آمدید
00:07	در این برنامه این موارد را توضیح می دهیم.
00:10	چگونه برنامه C++ را بنویسیم.
00:13	چگونه آن را کامپایل کنیم و چگونه آن را اجرا کنیم.
00:17	و تعدادی از اشتباهات رایج و راه حل آن را توضیح می دهیم
00:22	برای ضبط این برنامه من از Ubuntu operating system نسخه 11.10 و کامپایلر G++ نسخه 4/5/2 در Ubuntu
00:29	استفاده می کنم
00:35	برای تمرین این برنامه
00:37	شما باید با Ubuntu operating system و یک Editor آشنا باشید
00:44	بعضی از editor ها vim و gedit می باشند
00:48	من در اینجا از gedit استفاده میکنم
00:50	برای برنامه آموزشی های مربوطه لطفاً به وب سایت نشان داده شده مراجعه کنید
00:56	با یک مثال نوشتن یک برنامه C++ را نشان می دهیم
01:01	با فشار دادن همزمان کلید های Ctrl, Alt و T روی صفحه کلید یک پنجره ترمینال را باز کنید
01:09	برای باز کردن editor متنی در ترمینال gedit فاصله talk نقطه .cpp فاصله & را تایپ کنید
01:13	
01:21	ما از & برای آزاد سازی prompt استفاده می کنیم
01:25	تمامی فایل های C++ امتداد .cpp را دارند
01:31	حالا Enter را فشار دهید
01:33	Editor متنی باز شده است
01:35	حالا نوشتن برنامه را آغاز می کنیم
01:38	// فاصله My first c++ program را تایپ کنید
01:44	در اینجا double slash برای کامنت دادن به خط استفاده می شود
01:49	کامنتها برای فهمیدن جریان برنامه استفاده می شوند
01:52	این برای مستند مفید می باشد
01:55	و به ما در مورد برنامه اطلاعات می دهد.
01:59	Double slash کامنت یک خطی نامیده می شود. حالا Enter را فشار دهید
02:05	#Include فاصله کروشه زاویه ای باز و کروشه زاویه ای بسته را تایپ کنید
02:13	این یک تمرین خوب می باشد که در ابتدا کروشه را کامل کنید و سپس شروع به نوشتن در داخل آن کنید.
02:20	حالا بین کروشه ها iostream را تایپ کنید

02:23	lostream یک header file می باشد
02:26	این فایل شامل عملگرهای استاندارد ورودی و خروجی در C++ می باشد. حالا Enter را فشار دهید
02:35	Using فاصله namespace فاصله std و ; را تایپ کنید
02:45	بیانیه using به کامپایلر اطلاع می دهد که std namespace را می تواند استفاده کند
02:52	هدف از namespace جلوگیری از برخورد نام می باشد
02:56	و این با متمرکز کردن اسمهای مشخص کننده ها انجام می شود
03:00	این یک ناحیه ایجاد می کند و محدوده را تعریف می کند.
03:05	هر چیزی که بین namespace قرار بگیرد در محدوده (scope) آن namespace می باشد
03:11	در اینجا std همان namespace می باشد که در آن کل کتابخانه استاندارد C++ نشان داده شده. Enter را فشار دهید
03:20	Int فاصله main پراتنز باز پراتنز بسته را تایپ کنید
03:26	و main یک function خاص می باشد
03:30	این نشان می دهد که اجرای برنامه از این خط شروع می شود
03:34	به براکتهای باز و بسته پراتنز (parenthesis) گفته می شود
03:39	پراتنرهای بعد از main به کاربر می گوید که main یک function می باشد
03:45	در اینجا عملگر int main() هیچ arguments ندارد و به آن نوع عدد صحیح را می دهد.
03:52	ما در مورد نوع داده ها در برنامه دیگر آموزش می بینیم
03:56	حالا برای دانستن عملگر main به اسلایدهایمان برمی گردیم
04:02	هر برنامه باید یک عملگر main داشته باشد
04:04	نباید بیشتر از یک عملگر main داشته باشد
04:09	در غیر این صورت کامپایلر نمی تواند شروع برنامه را پیدا کند.
04:13	پراتنرهای خالی نشان می دهد که main هیچ arguments ندارد.
04:19	مفهوم arguments را در برنامه آموزشی بعدی توضیح می دهیم.
04:24	حالا به برنامه برمی گردیم. Enter را فشار دهید.
04:29	آکولاد راست را تایپ کنید. }
04:32	آکولاد راست شروع عملگر main را نشان می دهد.
04:37	و بعد آکولاد چپ را تایپ کنید. }
04:40	و آکولاد چپ پایان عملگر main را نشان می دهد.
04:45	حالا داخل آکولاد enter را دو بار فشار دهید.
04:49	نشانگر را یک خط بالاتر ببرید.
04:51	Indentation (دندانه دار کردن) کد را برای خواندن آسان تر می کند.
04:54	همچنین کمک می کند که اشتباه را سریع تر پیدا کنیم.
04:58	حالا در اینجا یک فاصله می دهیم.

05:01	و حالا cout فاصله دو براکت باز زاویه ای تایپ کنید
05:07	در اینجا cout یک C++ function استاندارد برای چاپ خروجی در ترمینال می باشد.
05:14	حالا بعد از براکت بین نقل قول دو گانه (double quotes) تایپ کنید.
05:18	هر چیزی بین double quotes در عملگر cout چاپ خواهد شد.
05:24	حالا بین نقل قول "n" Talk to a teacher backslash n را تایپ کنید.
05:31	در اینجا \n خط جدید را نشان می دهد.
05:35	در نتیجه بعد از اجرای عملگر cout نشانگر به خط جدید می رود.
05:41	هر دستور C++ باید به semicolon ختم شود.
05:45	پس آن را در آخر این خط تایپ کنید.
05:48	Semicolon بعنوان پایان دهنده دستور می باشد. حالا Enter را فشار دهید.
05:53	در اینجا فاصله دهید و return فاصله 0 و ; (semicolon) را تایپ کنید.
06:00	این بیانیه عدد صفر را می دهد.
06:03	عدد صحیح باید به این function داده شود چون نوع function در اینجا int می باشد.
06:10	دستور return پایان بیانیه قابل اجرا را نشان می دهد.
06:14	در مورد مقادیر return در برنامه آموزشی دیگر بیشتر یاد می گیریم.
06:20	برای ذخیره فایل دکمه Save را کلیک کنید.
06:23	عادت خوبی است که دائماً فایلها را ذخیره کنید.
06:26	این شما را از قطع ناگهانی برق محافظت می کند
06:30	و در صورت سقوط application نیز مفید می باشد.
06:34	حالا برنامه را کامپایل می کنیم.
06:36	به ترمینال خود باز می گردیم.
06:39	g++ فاصله talk.cpp فاصله -o فاصله output را تایپ کنید.
06:49	در اینجا g++ کامپایلری است که برای کامپایل برنامه های C++ استفاده می شود
06:55	نام فایل ما talk.cpp می باشد.
06:59	output -o Hyphen میگوید که قابل اجراها باید به فایل output برود.
07:05	حالا Enter را فشار دهید.
07:07	می بینیم که برنامه کامپایل می شود.
07:10	با تایپ کردن ls space hyphen lrt می بینیم که output آخرین فایلی است که باید ایجاد شود.
07:19	حالا برنامه را اجرا می کنیم ./output. را تایپ کنید.
07:24	و Enter را فشار دهید.
07:27	در اینجا خروجی بعنوان Talk to a teacher نمایش داده می شود.
07:30	حالا اشتباهات رایج را که ممکن است انجام دهیم می بینیم .

07:35	به ویراستار (editor) متنی خود باز می‌گردیم.
07:38	فرض کنید که در اینجا آکولاد چپ { را فراموش می‌کنیم.
07:42	فایل را ذخیره کنید.
07:44	حالا اجرا می‌کنیم به ترمینال خود باز می‌گردیم
07:48	حالا کامپایل و برنامه را با استفاده از command های قبلی اجرا می‌کنیم یک اشتباه می‌بینیم
07:55	در خط شماره 7 در فایل talk.cpp یک اشتباه که (اکولاد در آخر ورودی انتظار می‌رود) (expected curly bracket at the end of the input) می‌بینیم.
08:07	به ویراستار (editor) متنی خود باز می‌گردیم.
08:09	همان‌طور که قبلاً گفتیم آکولاد چپ پایان عملگر main را نشان می‌دهد.
08:14	پس دوباره آکولاد را اینجا وارد کنید و حالا فایل را ذخیره کنید.
08:19	و دوباره آن را اجرا کنید.
08:21	شما می‌توانید command هایی را که قبلاً وارد کرده‌اید را با استفاده از فلش رو به بالا به یاد بیاورید.
08:26	و این کاری است که حالا من کردم
08:30	بله انجام می‌شود.
08:32	حالا یک اشتباه رایج دیگر را نشان می‌دهم.
08:35	به ویراستار (editor) متنی خود باز می‌گردیم.
08:37	فرض کنید که در اینجا std را فراموش کرده ایم.
08:41	فایل را ذخیره کنید.
08:44	به ترمینال خود باز می‌گردیم.
08:46	حالا کامپایل می‌کنیم.
08:48	می‌بینیم در خط 3 و خط 6 در فایل talk.cpp اشتباهی وجود دارد.
08:56	که identifier قبل از semicolon ; انتظار می‌رود و cout در این محدوده اعلام نشده است.
09:05	چون cout یک C++ library function استاندارد می‌باشد.
09:09	کل C++ library function زیر std namespace تعریف می‌شود.
09:15	بنابراین اشتباه می‌دهد.
09:18	حالا اشتباه را برطرف می‌کنیم به ویراستار متنی خود باز می‌گردیم و در اینجا std را تایپ می‌کنیم.
09:23	آن را ذخیره می‌کنیم.
09:25	آن را دوباره کامپایل می‌کنیم.
09:29	بله اجرا می‌شود و حالا ارائه :
09:33	یک برنامه برای پرینت کردن نام خود و شهر خود بنویسید.
09:37	در این برنامه single line comment را استفاده کردیم.
09:40	حالا برای دادن multiline comment سعی کنید.
09:44	ویدیو را در لینک زیر مشاهده کنید.

09:46	این خلاصه پروژه Spoken Tutorial میباشد
09:48	اگر پهنای باند خوبی ندارید، ابتدا دانلود و سپس مشاهده کنید
09:53	تیم پروژه "Spoken Tutorial"
09:55	کارگاه آموزشی استفاده از "Spoken Tutorial" ارائه میدهد
09:58	و به کسانی که آزمون آنلاین را قبول شوند گواهینامه میدهد
10:01	برای جزئیات بیشتر لطفاً به "contact@spoken-tutorial.org" ایمیل بفرستید
10:10	Spoken tutorial بخشی از پروژه Talk To a Teacher می باشد.
10:14	که تحت پشتیبانی National Mission on Education از طریق ICT ، توسط MHRD دولت هند می باشد
10:20	اطلاعات بیشتر درلینک زیر موجود می باشد "spoken hyphen tutorial dot org slash NMEICT hyphen Intro"
10:25	این برنامه توسط شبیم اقبال خان ترجمه و صداگذاری شده است
10:28	با تشکر از شما