

Time	Narration
00:01	به spoken tutorial در Statics blocks خوش آمدید.
00:06	در این برنامه یاد می‌گیریم که: static blocks چیست.
00:10	چگونه static blocks را اعلام کنیم. و چگونه static blocks را استفاده کنیم.
00:16	در اینجا ما از Ubuntu 14.04 و JDK 1.7 و Eclipse 4.3.1 استفاده می‌کنیم.
00:26	برای این برنامه باید شما با اصول Java و Eclipse IDE آشنایی داشته باشید.
00:34	شما همچنین باید با instance variables, static variables و static methods در Java آشنایی داشته باشید.
00:43	اگر نه برای برنامه‌های Java مربوطه به لینک زیر مراجعه کنید.
00:48	حالا در مورد static blocks یاد می‌گیریم.
00:52	یک static block معمولاً برای شروع مقادیر static variables استفاده می‌شود.
00:59	static block با استفاده از کلیدواژه static اعلام می‌شود.
01:03	static blocks زمانی که class در حافظه لود شده است، اجرا می‌شود.
01:08	اگر برنامه شامل static blocks باشد آن‌ها قبل از constructors فراخوانی می‌شوند.
01:14	ما نمی‌توانیم به instance variables داخل static block دسترسی داشته باشیم.
01:19	حالا به Eclipse می‌رویم و پروژه جدید با نام StaticBlockDemo را ایجاد می‌کنیم.
01:26	داخل این پروژه ما classes لازم برای نشان دادن استفاده از static blocks را ایجاد می‌کنیم.
01:33	حالا src folder را کلیک راست کنید و New → Class را کلیک کنید.
01:38	نام class را StudentEnroll تایپ کنید و Enter را فشار دهید.
01:44	حالا این کد را برای نشان دادن StudentEnroll class تایپ کنید.
01:49	توجه کنید که دو static variables count و orgname می‌باشند.
01:54	حالا → Source را کلیک کنید و Generate Constructor using Fields را انتخاب کنید.
02:00	کلیدواژه super را از کد ایجاد شده حذف کنید.
02:04	ما می‌خواهیم که هر وقت constructor فراخوانی شد پیام نیز چاپ شود.
02:09	پس در این constructor کد زیر را برای چاپ کردن Constructor invoked تایپ کنید.

02:15	حالا متد <code>showData()</code> را به این <code>class</code> برای چاپ کردن مقادیر متغیرها اضافه می کنیم.
02:21	پس کد زیر را تایپ کنید.
02:23	حالا <code>static block</code> را برای شروع کردن مقادیر <code>count</code> و <code>orgname</code> اضافه می کنیم.
02:29	کد زیر را تایپ کنید.
02:32	متغیرهای <code>orgname</code> و <code>count</code> که <code>static variables</code> هستند.
02:36	این دسته از کد بین آکولاد که با کلیدواژه <code>static</code> پیشفرض شده اند <code>static block</code> هستند.
02:42	این <code>static block</code> مقادیر برای <code>count</code> و <code>orgname</code> را بعنوان 100 و IITM شروع می کند.
02:51	داخل این <code>static block</code> کد زیر را برای پرینت کردن <code>static block-1 is invoked</code> تایپ کنید.
02:58	ما حالا یک <code>class</code> دیگر که شامل <code>main method</code> می باشد اضافه می کنیم.
03:03	پس <code>default package</code> را کلیک راست کنید و <code>New→Class</code> را کلیک کنید و نام را <code>Demo</code> تایپ کنید.
03:11	در این <code>class</code> ما <code>main method</code> را خواهیم داشت.
03:15	پس <code>main</code> را تایپ کنید و <code>Ctrl+space</code> را برای ایجاد <code>main method</code> فشار دهید.
03:21	ما یک <code>object</code> از <code>StudntEnroll class</code> ایجاد می کنیم.
03:25	کد زیر را برای ایجاد <code>object s1</code> تایپ کنید.
03:29	حالا <code>showData method</code> را برای چاپ کردن مقادیر فراخوانی می کنیم.
03:33	و <code>semicolon</code> <code>s1.showData()</code> را تایپ کنید.
03:38	حالا برنامه <code>Demo</code> را <code>run</code> می کنیم.
03:41	می بینیم که <code>static block</code> قبل از <code>constructor</code> فراخوانی شده است.
03:46	مقادیر <code>count</code> و <code>orgname</code> همانطور که در <code>static block</code> تعریف شده است، شروع می شوند.
03:53	حالا به <code>StudentEnroll class</code> برمی گردیم.
03:57	حالا ببینیم اگر مقدار <code>id</code> در <code>static block</code> را شروع کنیم چه می شود.
04:03	در <code>static block</code> شما <code>id equals IT01</code> <code>semicolon</code> را تایپ کنید.
04:10	می بینیم که اشتباه می دهد.
04:13	این نشان می دهد که به <code>instance variable</code> نمی توان دسترسی پیدا کرد.
04:19	حالا این خط را کامنت می دهیم.
04:25	به اسلایدهایمان برمی گردیم.
04:27	یک <code>class</code> می تواند شامل چندین <code>static blocks</code> باشد.

04:30	در این صورت آن‌ها را فقط یکبار به ترتیبی که در source کد ظاهر می‌شوند فراخوانی می‌کنیم.
04:37	برای بررسی این به Eclipse بروید.
04:40	حالا یک static block دیگر بعد از قبلی ایجاد می‌کنیم.
04:45	کد زیر را تایپ کنید.
04:47	این static block مقادیر برای count و orgname را بعنوان 200 و IITB شروع می‌کند.
04:57	داخل این static block کد زیر را تایپ کنید.
05:01	دوباره برنامه Demo را run می‌کنیم.
05:04	بعد از بررسی خروجی می‌بینیم که static block دوم بعد از اولی فراخوانی می‌شود.
05:10	مقادیر static variables count و orgname با static block دوم بروزرسانی می‌شود.
05:18	آن‌ها 200 و IITB هستند.
05:22	خلاصه می‌کنیم.
05:24	در این برنامه یاد گرفتیم که static block چیست. چگونه یک static block را اعلام و تعریف کنیم و چگونه static blocks را فراخوانی و اجرا کنیم.
05:37	ارائه: این ارائه ادامه ارائه Static Methods می‌باشد.
05:44	مطمئن شوید که ارائه Static Methods را کامل کرده اید.
05:48	مثل قبل class CarService را همانطور که در این اسلاید مشخص شده طراحی کنید.
05:54	و instance variables و static variables را شناسایی کنید.
05:58	Constructor را برای شروع کردن مقادیر برای instance variables تعریف کنید.
06:03	static block را برای شروع کردن مقادیر static variables تعریف کنید.
06:08	همچنین Demo class که شامل main method است را ایجاد کنید.
06:12	در main method تعدادی objects از CarService را ایجاد کنید و method show() را فراخوانی کنید.
06:19	ویدئو در لینک زیر خلاصه Spoken Tutorial Project می‌باشد،
06:24	اگر به‌نای باند خوبی ندارید ابتدا دانلود و سپس مشاهده کنید.
06:27	تیم این پروژه کارگاه آموزشی استفاده از Spoken Tutorial ارائه می‌دهد. و به کسانی که آزمون آنلاین را قبول شوند، گواهینامه می‌دهد. برای جزئیات بیشتر لطفاً contact AT spoken HYPHEN tutorial DOT org ایمیل بفرستید

06:35	برای جزئیات بیشتر لطفاً به ما ایمیل بفرستید .
06:39	بودجه Spoken Tutorial Project توسط NMEICT, MHRD دولت هند تأمین می شود.
06:46	اطلاعات بیشتر در لینک نشان داده شده در دسترس می باشد.
06:50	این script توسط Department of Information Technology, Amal Jyothi College of Engineering منتشر شده است.
06:58	ترجمه و صداگذاری شبیم اقبال خان. با تشکر از شما