

## به نام خدا

### آموزش بیست و دوم

اگر به حدوداً ۴۰ تا ۵۰ سال پیش بازگردیم، می بینیم که در آن دوران در مدارس روال به این شکل بود که برای تعطیلات نوروز دانش آموزان را مجبور می کردند تا به طور مثال از روی کتاب فارسی خود چند بار بنویسند چرا که در آن زمان پیک شادی وجود نداشت. در واقع دانش آموزان موظف بودند چیزی تکراری را بارها و بارها بنویسند تا مثلاً تعداد ۷ سری از نوشته ها تکمیل شود. حال پس از آنکه ۷ سری نوشتن از روی کتاب فارسی تکمیل شد از مغز دانش آموز دستور می رسد که کار به اتمال رسیده است و بازنویسی می بایست متوقف گردد.

در زبان برنامه نویسی به این کار Loop گفته می شود (لازم به ذکر است به جای آنکه معنی فارسی واژه Loop به معنی "حلقه" را حفظ کنیم، بهتر است بدون توجه به معنی فارسی آن، دستوری که این کار تحت آن انجام می شود را Loop بنامیم).

به عبارت دیگر Loop دستوری است که منجر می شود چیزی ثابت برای تعداد دفعاتی تعریف شده و یا تا بی نهایت تکرار شود. در زبان برنامه نویسی جاوا سه دستور برای استفاده از Loop ها وجود دارد که عبارتند از for، while، do که می توان هر یک از آنها را به جای دیگری مورد استفاده قرار داد اما نکته ای که می بایست مد نظر قرار داده شود این است که هر کدام از این دستورات می تواند بسته به اینکه چه انتظاری از Loop داریم کار ما را راحت تر کند.

### نحوه به کار گیری for

حال ببینیم که با دستورات Loop در برنامه چه کارهایی می توان انجام داد. در واقع زمانیکه آنتی ویروس نصب شده روی کامپیوتر یا لپ تاپ ما هر زمانیکه یک ایمیل جدید می آید آن را چک می کند تا ببیند حاوی ویروس است یا خیر، این کار توسط دستورات Loop انجام می شود. و یا زمانیکه محافظ صفحه نمایش کاری تکراری را بارها و بارها انجام می دهد (به طور مثال یک توپ مکرراً از سمت چپ به سمت راست می رود و بالعکس) این کار توسط Loop انجام می شود. در حقیقت دستورات Loop در یک برنامه کامپیوتری، برنامه را مجبور می کنند تا برای

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

تعداد دفعاتی که ما برای آن تعریف می کنیم به قسمت خاصی از برنامه بازگردند. برای روشن تر شدن این مطلب می توانیم بازی یویو را در نظر بگیریم که مثلاً ما به عنوان کسی که اختیار این بازی را در دست داریم به یویو دستور می دهیم تا به نقطه پیشین که در آن قرار داشت بارها و بارها بازگردد.

حال برای آنکه به طور عملی با Loop یی از جنس for آشنا شویم پروژه ای جدید تحت عنوان 22<sup>nd</sup> Session به معنی "جلسه بیست و دوم" ایجاد کرده و کلاسی تحت عنوان ForLoop در آن ایجاد می کنیم:

```
public class ForLoop {  
    public static void main(String[] args) {  
  
    }  
}
```

حال برای آنکه بتوانیم از Loop یی از جنس for استفاده کنیم، می بایست دستور {}for() را به کد خود اضافه کنیم (توجه داشته باشیم که حرف اول for به صورت کوچک نوشته شده است). حال داخل پرانتز مقابل for می بایست سه پارامتر را بگنجانیم به این شکل که پارامتر اول می بایست نقطه آغازین Loop ما باشد و پارامتر دوم می بایست نقطه انتهایی Loop باشد و به عبارت دیگر می توان پارامتر دوم را یک شرط در نظر گرفت که مادامیکه true است Loop ادامه خواهد یافت و به محض اینکه false شود برنامه از Loop خارج خواهد شد و در نهایت در پارامتر سوم می بایست مشخص کنیم که مقدار پارامتر نقطه آغازین در هر بار Loop شدن به چه مقدار افزایش یابد. حال در این مثال قصد داریم تا به برنامه خود دستور دهیم که از نقطه آغازین ۱ شروع کرده و پس از هر بار Loop شدن یک واحد به آن اضافه کند و در نهایت در نقطه پایانی ۱۰ توقف نماید. کد تکمیل شده به شکل زیر خواهد بود:

```

public class ForLoop {
    public static void main(String[] args) {
        for (int number = 1; number <= 10; number++) {
            System.out.println(number);
        }
    }
}

```

در کد فوق یک متغیر از جنس `int` که در برگیرنده اعداد صحیح خواهد بود تحت عنوان `number` به معنی "**عدد**" با مقدار اولیه 1 برای نقطه آغازین تعریف کرده ایم سپس یک علامت `<=` قرار می دهیم و بلافاصله نقطه پایانی را برای `Loop` خود در نظر می گیریم که عبارت است از `number <= 10` به این معنی که این `Loop` ادامه خواهد یافت تا زمانی که `Loop` ما به عدد 10 برسد. علت آنکه علامت کوچکتر مساوی گذاشته ایم این است که خود عدد 10 را هم شامل شود که در غیر این صورت مادامیکه `Loop` کوچکتر از 10 بود ادامه می یافت. به عبارت دیگر زمانی که `Loop` به عدد 9 می رسید توقف می کرد زیرا که شرط قرار گرفته در `Loop` برآورده می شد. در نهایت مقداری که می خواهیم متغیر ما در هر دفعه `Loop` شدن افزایش یابد را در نظر می گیریم که در این مثال یک واحد است (برای آشنایی بیشتر با اعمال ریاضیاتی به آموزش `Operator` ها در قسمت هشتم مراجعه نمایید). حال برای آنکه `Loop` را روی مانیتور به نمایش در آوریم، نام متغیر خود را در دستور `System.out.println()` قرار می دهیم. نکته ای که در مورد کد فوق وجود دارد این است که می توان متغیر را خارج از `for` نیز تعریف کرد. به عبارت دیگر کد فوق را به شکل زیر نیز می توان بازنویسی نمود:

```

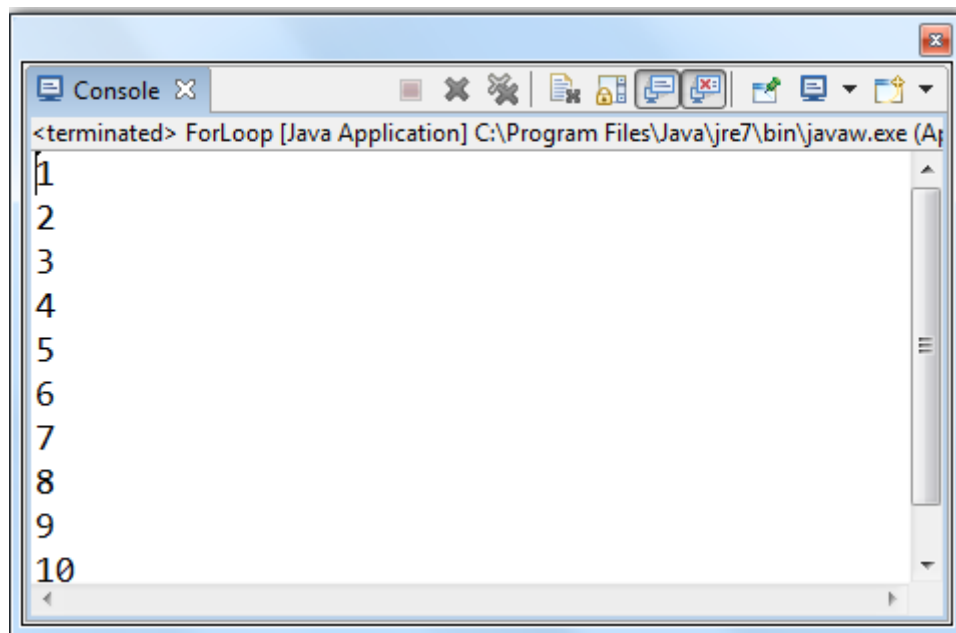
public class ForLoop {
    public static void main(String[] args) {
        int number;
        for (number = 1; number <= 10; number++) {
            System.out.println(number);
        }
    }
}

```

در واقع ما یک متغیر از جنس `int` خارج از `Loop` ایجاد کرده ایم، حال داخل `for` مقدار اولیه، مقدار نهایی و همچنین مقدار افزایشی برای آن تعریف می کنیم.

پس از اجرای برنامه خواهیم دید که `Loop` ما از عدد یک شروع شده و به عدد ده خاتمه می

یابد:



اکنون برای آنکه مفهوم `for` برای ما نهادینه تر شود شروع می کنیم به اعمال یکسری تغییر در دستور `for` تا اینکه ببینیم چه نتیجه ای در بر خواهد داشت. فرض کنیم که می خواهیم برنامه ای طراحی کنیم که از عدد یک شروع می کند به شمردن و سه تا سه تا افزایش می یابد تا به عدد بیست و دو برسد. برای این منظور کد فوق را به شکل زیر بازنویسی می کنیم:

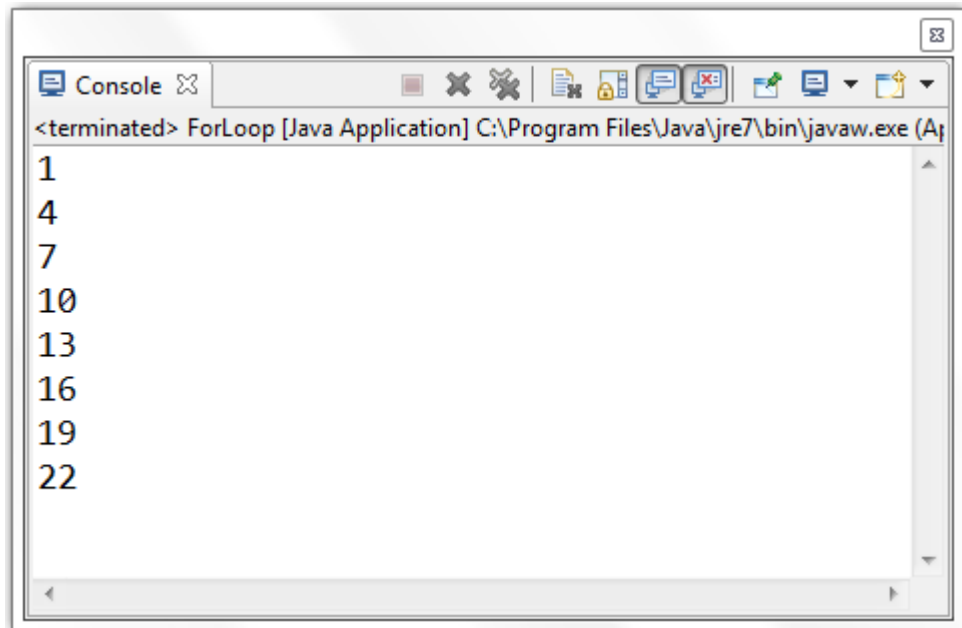
```
public class ForLoop {  
    public static void main(String[] args) {  
        for (int number = 1; number <= 22; number += 3) {  
            System.out.println(number);  
        }  
    }  
}
```

خروجی برنامه جدید به شکل زیر خواهد بود:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی



همانطور که مشاهده می شود Loop ما کار خود را از عدد یک شروع می کند. همانطور که در آخرین پارامتر دستور داده ایم، در هر بار Loop شدن می بایست سه واحد به واحد قبلی اضافه شود، از این رو در بار دوم Loop شدن سه واحد به واحد یک اضافه شده مقدار متغیر به چهار افزایش می یابد، در بار بعد سه واحد به عدد چهار اضافه می شود و به هفت افزایش می یابد الی آخر.

حال فرض کنیم که می خواهیم برنامه خود را به نوعی بازنویسی کنیم که تا بی نهایت شروع به شمردن کند. برای این منظور کد خود را به شکل زیر بازنویسی خواهیم کرد:

```
public class ForLoop {  
    public static void main(String[] args) {  
        for (int number = 1; number > 0; number++) {  
            System.out.println(number);  
        }  
    }  
}
```

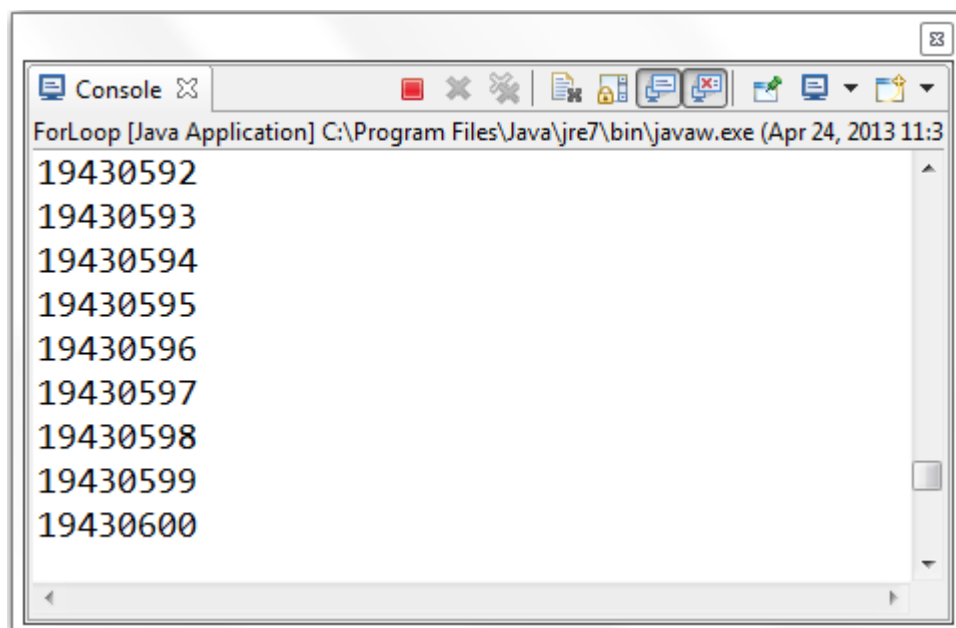
در این کد مقدار اولیه را یک در نظر گرفته ایم، سپس به Loop دستور داده ایم تا زمانی که مقدار متغیر number بزرگتر از صفر است در هر بار Loop شدن یک واحد به آن اضافه کند. در واقع از آنجا که مقدار اولیه متغیر ما یک بوده و منطقاً عدد یک از صفر بزرگتر است پس هیچ نقطه پایانی هم برای آن در نظر گرفته نشده است، Loop ما تا بی نهایت ادامه خواهد داشت و در هر

دوره آموزش جاوا

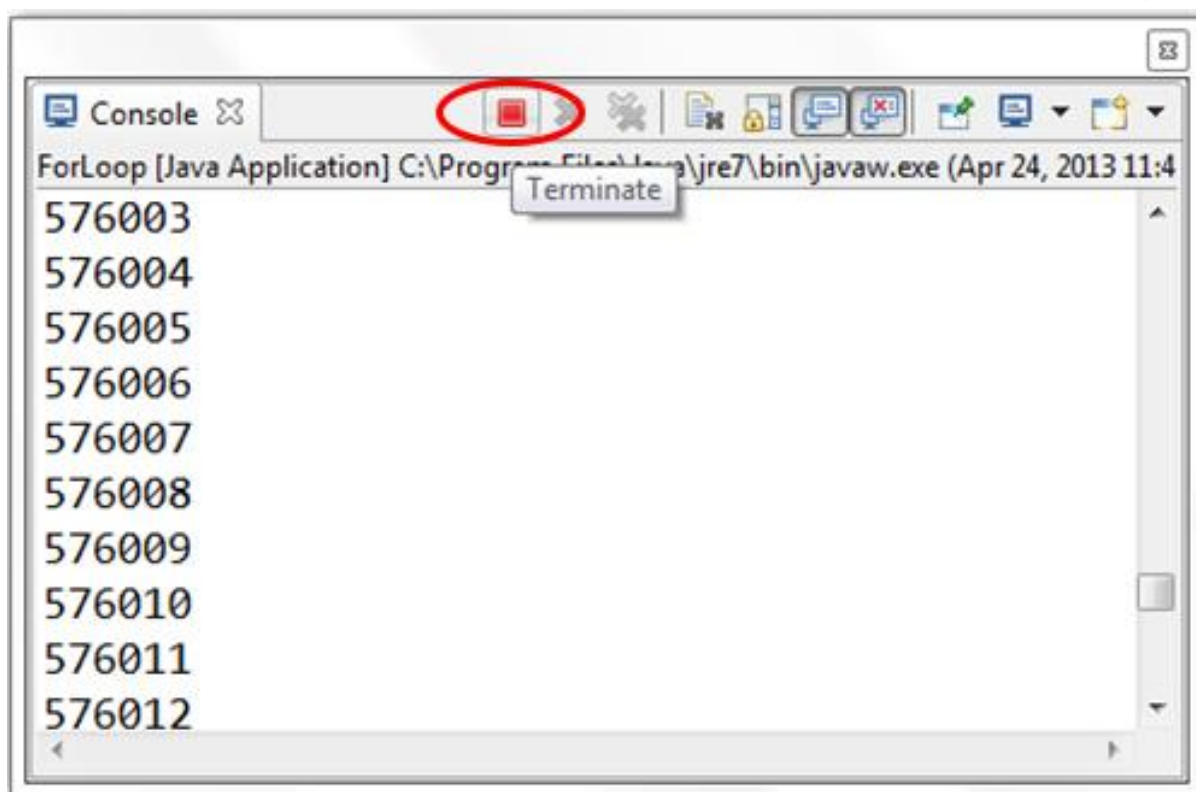
کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

بار شدن یک واحد به آن اضافه خواهد شد. در واقع خروجی برنامه به شکل زیر خواهد بود:



در واقع این Loop که ایجاد کردیم تا بی نهایت ادامه خواهد یافت تا اینکه ما خود آن را متوقف نماییم:



دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

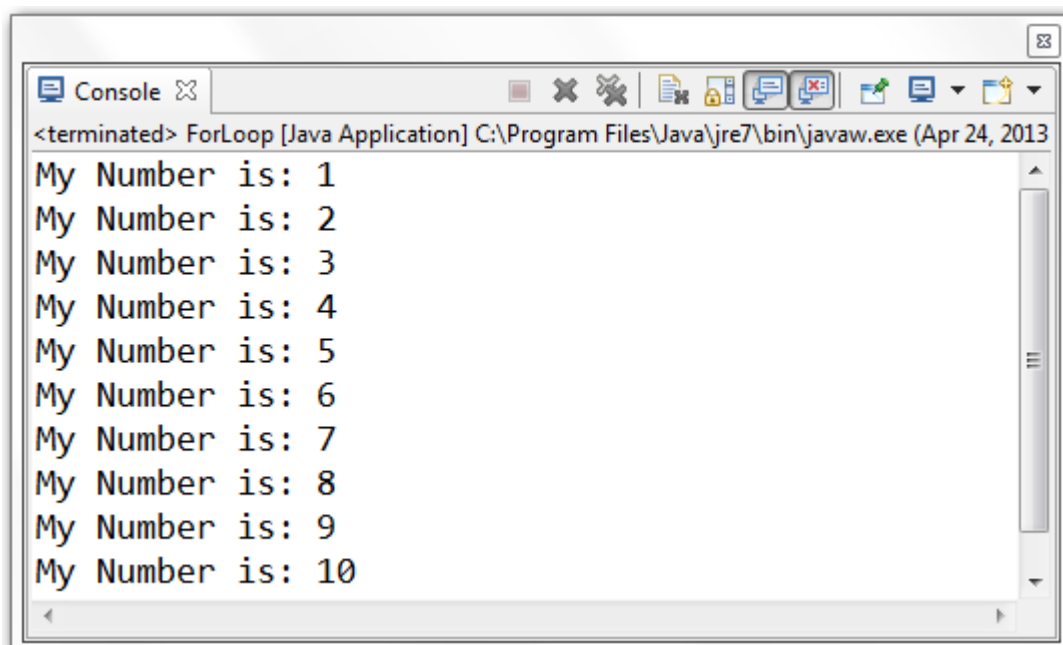
که این کار با فشردن دکمه قرمز رنگ در بخش Console تحت عنوان Terminate عملی خواهد شد که در تصویر فوق با یک بیضی قرمز رنگ مشخص شده است.

حال فرض کنیم می خواهیم خروجی Loop را با یک string یا عبارت ادغام نماییم. برای این منظور به شکل زیر کد خود را بازنویسی می کنیم:

```
public class ForLoop {  
    public static void main(String[] args) {  
        for (int number = 1; number <= 10; number++) {  
            System.out.println("My Number Is: " + number);  
        }  
    }  
}
```

در Statement یی که در آن دستور داده ایم تا نوشته ای روی صفحه مانیتور چاپ شود که همان System.out.println(); می باشد یک string تحت عنوان My number Is: به معنی "شماره من هست:" نوشته ایم و بلافاصله بعد از آن از یک Operator به علاوه استفاده کرده ایم تا string خود را به یک متغیر وصل کنیم که در اینجا همان متغیر number است.

حال خروجی برنامه به شکل زیر خواهد بود:

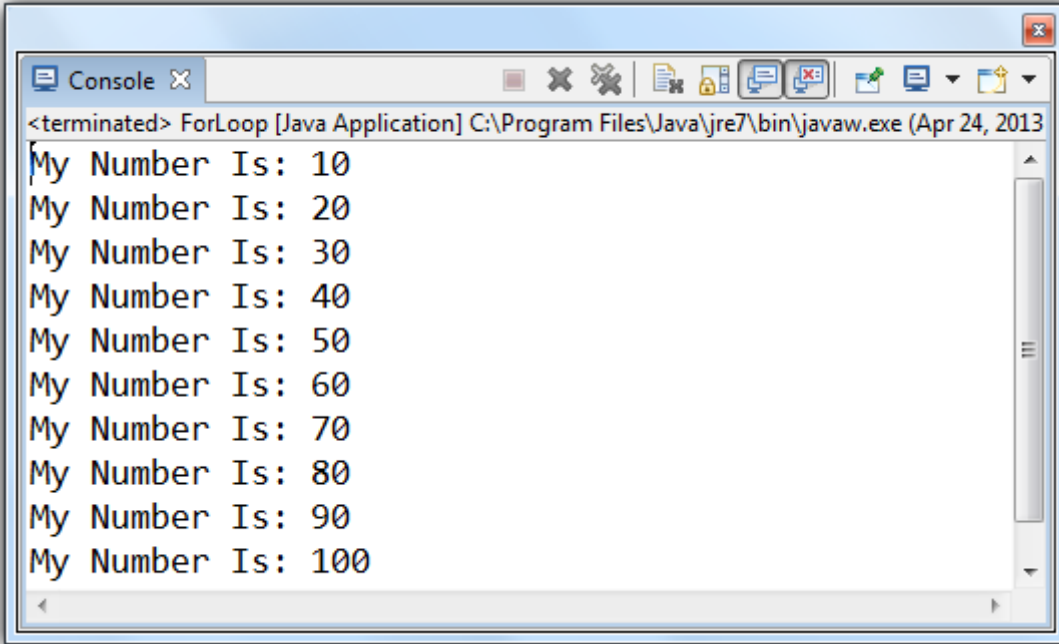


```
<terminated> ForLoop [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Apr 24, 2013  
My Number is: 1  
My Number is: 2  
My Number is: 3  
My Number is: 4  
My Number is: 5  
My Number is: 6  
My Number is: 7  
My Number is: 8  
My Number is: 9  
My Number is: 10
```

اکنون فرض کنیم که هر دفعه که قرار است برنامه یک Loop انجام دهد می خواهیم تا جواب Loop در عدد ۱۰ ضرب شود سپس حاصلضرب خروجی Loop در عدد ۱۰ در بخش Console به نمایش در آید. برای همین منظور کد خود را به شکل زیر بازنویسی می کنیم:

```
public class ForLoop {  
    public static void main(String[] args) {  
        for (int number = 1; number <= 10; number++) {  
            int multiply = number * 10;  
            System.out.println("My Number Is: " + multiply);  
        }  
    }  
}
```

در این کد ما یک متغیر جدید از جنس int ایجاد کرده ایم تحت عنوان multiply به معنی "ضرب کردن" و مقدار آن را عبارت است از هرآنچه مقدار متغیر number اختصاص داده شده به for بود، ضرب در عدد ۱۰ و در نهایت برای آنکه مقدار متغیر جدید ما در محیط Console به نمایش در آید در دستور System.out.println(); به جای نام متغیر قبلی، نام متغیر جدید که همان multiply است را قرار می دهیم که در این صورت خروجی برنامه به شکل زیر خواهد بود:

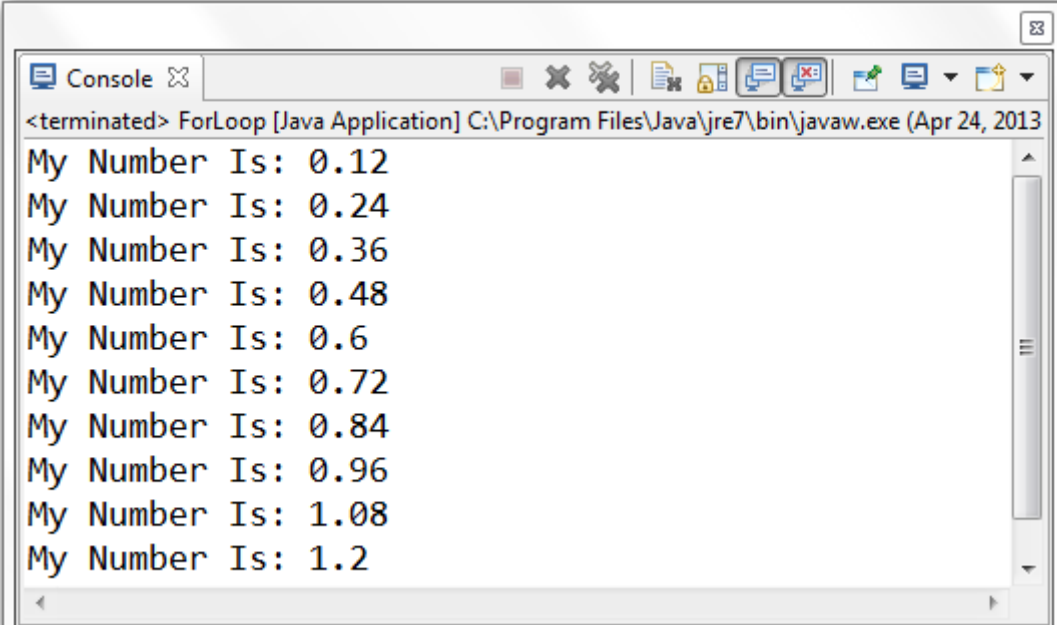




به عنوان مثال آخر، این نکته را نیز می بایست مد نظر قرار داده که به جای متغیری از جنس `int` که برای اعداد صحیح است، می توان از `double` هم برای استفاده از اعداد اعشاری هم استفاده کرد. فرض کنیم که می خواهیم برنامه خود را به این شکل تغییر دهیم که مقدار متغیر اختصاص داده شده به `for` پس از آنکه `Loop` شد در عدد دوازده صدم ضرب شود. برای این منظور کد خود را به شکل زیر بازنویسی می کنیم:

```
public class ForLoop {
    public static void main(String[] args) {
        for (int number = 1; number <= 10; number++) {
            double multiply = number * 0.12;
            System.out.println("My Number Is: " + multiply);
        }
    }
}
```

در واقع در این کد تنها تغییری که رخ داده این است که جنس متغیر `multiply` از `int` به `double` تغییر یافته و همچنین عددی که می بایست مقدار متغیر `number` در آن ضرب شود از عدد ده به دوازده صدم تغییر پیدا کرده است. حال می توانیم انتظار داشته باشیم که خروجی برنامه ما به شکل زیر تغییر یابد:



```
<terminated> ForLoop [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Apr 24, 2013)
My Number Is: 0.12
My Number Is: 0.24
My Number Is: 0.36
My Number Is: 0.48
My Number Is: 0.6
My Number Is: 0.72
My Number Is: 0.84
My Number Is: 0.96
My Number Is: 1.08
My Number Is: 1.2
```

پس از آشنایی با Loop یی از جنس for در جلسه آموزشی آینده به بررسی دیگر انواع Loop ها همچون while خواهیم پرداخت.