

(لوگو) جو (مال و متاع) تم کو دیا گیا ہے وہ دنیا کی زندگی کا (ناپائیدار) فائدہ ہے۔ اور جو کچھ خدا کے ہاں ہے وہ بہتر اور قائم رہنے والا ہے

(یعنی) ان لوگوں کے لئے جو ایمان لائے اور اپنے پروردگار پر بھروسہ رکھتے ہیں۔ الشوری آیت 36

We are all born ignorant, but one must work hard to remain stupid. Benjamin Franklin

Lecture 11

Lecture 11 FOR Loop Syntax and Loop Combined with Selection

As I already discussed we are passing through the critical stages of learning in programming. Today we will see syntax of for loop and discuss some scenarios where selection becomes part of repetition and finally we will see example where repetition becomes part of selection. So let's start to learn something new and interesting.

For Loop

There is nothing special about for loop except syntax which seems to be relatively simpler for coding. Just recall counting loop has three essential elements initialization, condition and increment. Syntax of for loop provides a specific place to put all 3 elements, therefore, for loop is a natural candidate for counting loops; however, any while loop can be written using for loop because syntax of for loop allows same condition as while loop. The syntax of for loop is:

```
for ( stmt ; cond ; stmt )
```

example is:

```
for (i=1 ; i<=10 ; i=i+1)
```

Just compare a program written using both while and for syntax and try to understand the small difference only in use:

<pre>i=1; while (i<=10){ System.out.print(i+" "); i=i+1; }</pre>	<pre>for (i=1;i<=10;i=i+1){ System.out.print(i+" "); } for (i=1;i<=10; i=i+1) System.out.print(i+" ");</pre>
---	--

All these programs have same output that is 1 2 3 ... 10 in a single line. On right hand side two versions of for syntax is given, because there is only one line after for statement, therefore, curly braces are optional. See another example:

<pre>System.out.print ("Enter Table No:"); t=in.nextInt(); for (i=1;i<=10; i=i+1) System.out.println(t+"\t*\t"+i+"\t=\t"+(t*i));</pre>

It is very simple program to generate table of user input number. See another example:

<pre>String space=" "; for (i=0,j=10;i<6;i++){ System.out.print(space.substring(0,i)+"*"); System.out.println(+space.substring(0,j)+"\b*"); j=j-2; }</pre>	<pre>* * * * * * * * * * * *</pre>
--	--

This program is creating V with stars. There are two space patterns used. One is before first star on each row and second is between first and second star in each row. First pattern is increasing by

one; whereas; second pattern is decreasing by 2. Therefore two variables are used "i" for first pattern and "j" for second pattern. "i" is incremented by one and "j" is decremented by 2. Lastly \b is used to erase last character before *. This helps to erase one star in last line, so you can see there is only one star in last row of output given on right hand side.

Repetition with Selection

Now we will see that selection can be a part of repetition, therefore, each time we can take some decision in repetition. Consider scenario we have English word and we want to show about each character whether it is vowel or consonant. We already did examples where we can show each character in single line using function of String class. See again that code before I jump to required code:

```
String s="wonderful";
for (i=0;i<s.length(); i=i+1)
    System.out.println(s.charAt(i));
```

Now we want to print information about type of each character, because each character can be vowel or consonant we have to apply selection that if character is equal to any of the five vowel characters print character is vowel otherwise it is consonant because no other choice is there. Now writing if (s.charAt(i)=='a' || s.charAt(i)=='e' ... is inconvenient, therefore, it is better to use a character type variable to store each character than compare like:

```
ch=s.charAt(i);
```

```
if (ch=='a' || ch=='e' ...
```

Finally we can break print into two steps, because for each character we have to print that character and " is ", therefore we can write one print statement without next line option before if statement and one print statement with next line option with if statement. Finally here is code:

<pre>String s="wonderful"; char ch; for (i=0;i<s.length(); i=i+1){ ch=s.charAt(i); System.out.print(s.charAt(i)+" is "); if (ch=='a' ch=='e' ch=='i' ch=='o' ch=='u') System.out.println("vowel"); else System.out.println("consonant"); }</pre>	<pre>w is consonant o is vowel n is consonant d is consonant e is vowel r is consonant f is consonant u is vowel l is consonant</pre>
---	---

Hope you have enjoyed the example and understand selection inside repetition, however, we will discuss more examples. Suppose for numbers 40 to 59 we want to show are they divisible by 2, 3, 4, ... , 9. That is for each number we want to show its divisors among 2 to 9. Therefore, using repetition we can print numbers 40 to 59 but to show divisors we have to apply selection so that if number is divisible than we show divisor. Like if (i%2==0) or if (i%3==0) and so on. Therefore we will place selection inside repetition. Moreover because all of these checks are independent to each other we have independent selection statements. So here is code:

<pre> for (i=40;i<60;i++){ System.out.print(i+" is divisible by ..."); if (i%2==0) System.out.print(2+","); if (i%3==0) System.out.print(3+","); if (i%4==0) System.out.print(4+","); if (i%5==0) System.out.print(5+","); if (i%6==0) System.out.print(6+","); if (i%7==0) System.out.print(7+","); if (i%8==0) System.out.print(8+","); if (i%9==0) System.out.print(9+","); System.out.println("\b."); } </pre>	<p>40 is divisible by ...2,4,5,8. 41 is divisible by ... 42 is divisible by ...2,3,6,7. 43 is divisible by ... 44 is divisible by ...2,4. 45 is divisible by ...3,5,9. 46 is divisible by ...2. 47 is divisible by ... 48 is divisible by ...2,3,4,6,8. 49 is divisible by ...7. 50 is divisible by ...2,5. 51 is divisible by ...3. 52 is divisible by ...2,4. 53 is divisible by ... 54 is divisible by ...2,3,6,9. 55 is divisible by ...5. 56 is divisible by ...2,4,7,8. 57 is divisible by ...3. 58 is divisible by ...2. 59 is divisible by ...</p>
---	---

Once again we have used `\b` to erase last comma sign. This program can be improved further by showing no divisor etc. but that will program complex for this stage. Last example is to print 20 to 99 in words that is twenty to ninety nine. We intentionally selected this range because there are many other checks required for counting of 1 to 100. The technique we used is to separate first and second digit of each number that is in 38 first digit is 3 and second digit is 8. For this handy tool is integer division and remainder operator. That is dividing number by 10 we will get first digit and by taking remainder by 10 we will get second digit:

```
fDigit=76/10;
sDigit=76%10;
```

Here `fDigit` means first digit will be 7 and `sDigit` means second digit will be 6. Further we applied checks on first digit and assigned value Twenty to Ninety to string variable first than apply check on second digit and assigned value One to Nine to string variable second. Finally we print both first and second. We place all this inside loop and run loop from 20 to 99. Here is code:

<pre> int fDigit, sDigit; for (i=20;i<=99;i++){ fDigit=i/10; sDigit=i%10; String first="", second=""; if (fDigit==2) first="Twenty "; else if (fDigit==3) first="Thirty "; else if (fDigit==4) first="Fourty "; ... else if (fDigit==9) first="Ninty "; if (sDigit==1) </pre>	<p>Twenty Twenty One Twenty Two Twenty Three Twenty Four Twenty Five Twenty Six Twenty Seven Twenty Eight Twenty Nine Thirty Thirty One Thirty Two Thirty Three Thirty Four Thirty Five Thirty Six Thirty Seven Thirty Eight Thirty Nine Fourty Fourty One Fourty Two Fourty Three Fourty Four Fourty Five Fourty Six Fourty Seven Fourty Eight Fourty Nine Fifty Fifty One Fifty Two Fifty Three Fifty Four Fifty Five Fifty Six Fifty Seven Fifty Eight Fifty Nine Sixty Sixty One Sixty Two Sixty Three Sixty Four Sixty Five Sixty Six Sixty Seven Sixty Eight Sixty Nine Seventy Seventy One Seventy Two Seventy Three Seventy Four</p>
--	---

<pre> second="One"; else if (sDigit==2) second="Two"; else if (sDigit==3) second="Three"; ... else if (sDigit==9) second="Nine"; System.out.print(first+second+"\t"); } </pre>	<pre> Seventy Five Seventy Six Seventy Seven Seventy Eight Seventy Nine Eighty Eighty One Eighty Two Eighty Three Eighty Four Eighty Five Eighty Six Eighty Seven Eighty Eight Eighty Nine Ninty Ninty One Ninty Two Ninty Three Ninty Four Ninty Five Ninty Six Ninty Seven Ninty Eight Ninty Nine </pre>
--	---

Repetition inside Selection

A last thing for today's lecture is to discuss scenarios where repetition can become part of selection. Just consider that we want to print number from some starting value to end value and start value and end values are given by user as input. This is very simple to handle like:

```

int start = in.nextInt();
int end = in.nextInt();
for ( ; start <= end ; start = start +1 )
    System.out.println(start);

```

Problem is when we ask user to give input we can expect any blunder from user and somehow we can handle some of them like in this case we can write two loops one is from start to end and other is from end to start or we can also write both start to end but one is increasing and other is decreasing. Consider code:

<pre> int start, end; Scanner in = new Scanner (System.in); System.out.print("Enter Starting Number:"); start = in.nextInt(); System.out.print("Enter Ending Number:"); end = in.nextInt(); if (start<=end) while (start<=end){ System.out.println(start); start=start+1; } else while (end<=start){ System.out.println(end); end=end+1; } </pre>	<pre> if (start<=end) while (start<=end){ System.out.println(start); start=start+1; } else while (start>=end){ System.out.println(start); start =start-1; } </pre>
--	---

Here we have given two codes, one on left hand side and second on right hand side; whereas first 6 lines are common for both. In left hand side code we have 2 repetitions one from start to end and second from end to start depending on whether user give start less than end or otherwise. Whereas in right hand side code, we have both loop from start to end but first in incrementing towards end whereas second is decrementing on towards end. Lastly output of first code will be same if user replaces start and end; whereas in second code output will be reversed.

For further practice see lab 6 coming soon on website InshaAllah.