(مسلمانو) سب نمازیں خصوصاً بیچ کی نماز (یعنی نماز عصر) پورے التزام کے ساتھ ادا کرتے رہو۔ اور خدا کے آگے

ادب سے کھڑے رہا کرو ۔البقرۃ آیت ۲۳۸

We are what we repeatedly do. Excellence, then, is not an act, but a habit. Aristotle

Lecture 09

Practice using Selection & Intro to Repetition

So far I am not getting feedback about these lecture notes. Kindly share if you have any suggestion or positive critics.

In this lecture we start with some further practice of selection and later we will start an introduction to repetition.

Practice using Selection

Suppose you have student id stored in String type variable:

```
String sId1="MITF09M012", sId2="BSEF07A012";
```

You can do many queries from these ID's:

- **Students degree:**

```
if (sID1.substring(0,3).equals("MIT"))
    ? ("Masters in Information Technology");
else if (sID1.substring(0,3).equals("BSE"))
    ? ("Bachelor of Software Engineering");
...
```

- **Students Year of Admission:**

```
if (sID1.substring(4,6).equals("09"))
    ? ("2009");
else if (sID1.substring(4, 6).equals("10"))
    ? ("2010");
...
```

- **Students Section:**

```
if (sID1.charAt(6)== 'M ')
    ? ("Morning");
else
    ? ("Afternoon");
```

- **Is Students Enrolled in same year:**

```
if (sID1.substring(4,6).equals(sID2.substring(4,6)))
    ? ("Yes");
else
    ? ("No");
```

REPETITION

Repetition means to execute statement or block of statements number of time. Normally each statement execute once in a sequence. Using if condition we can select or reject a statement. Students must understand if condition is executed only and only once after that code proceeds to next statements or terminates in any case. There is no way **if** condition can be executed multiple times unless if statement is in loop. For example if I write:

```
int count=1;
if (count < 10){
    ? (count);
    count=count+1;
}
? (count);
```

What will be the output of this code? Surely it is **[1   2]**, because if condition evaluates to true therefore count will be printed once inside if body and count will be incremented by 1. After the line count=count+1, if condition may be true but it is important to understand that if condition will not be executed again, control
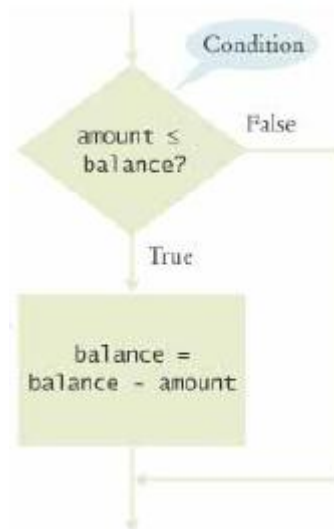
will be transferred to next statement that is ? (count) outside curly braces. See flow of **if** statement in flow chart. Condition box has a condition to check balance. If amount is less than equal to balance then amount can be drawn otherwise not, but you can notice in either case next statement will be executed in both cases; where you can see both arrows joining.

Figure taken from Big Java 3rd Edition by Cay Hortsman

Understanding the behavior of If statement, now we will discuss repetition. In repetition we have some condition like **if** but here we will check condition again and again till it remains true; and each time condition is true statement or block of statements also executes. When condition goes false control move to statement next to repetition statements.

Flow of repetition is shown in figure. One can see condition somehow similar to previous one and apparently flow is similar but flow at the end is different. If condition is true then flow ultimately comes back to condition; whereas in case of false flow moves below the repetition structure.

Here condition is "Is balance less than targetBalance". If condition evaluates to true, control goes to next block that is "Increment Years" than control goes to next block that is "Add interest to balance" than flow   goes back to condition and it will continue like that till condition evaluates to true and lastly when condition evaluates to false flow goes out of loop.

Once again difference between selection and loop is a selection is executed only once whereas loop can run infinite times. See following pseudo code:

```
counter = 1
while counter <>2
    print counter
    counter = counter + 2
```
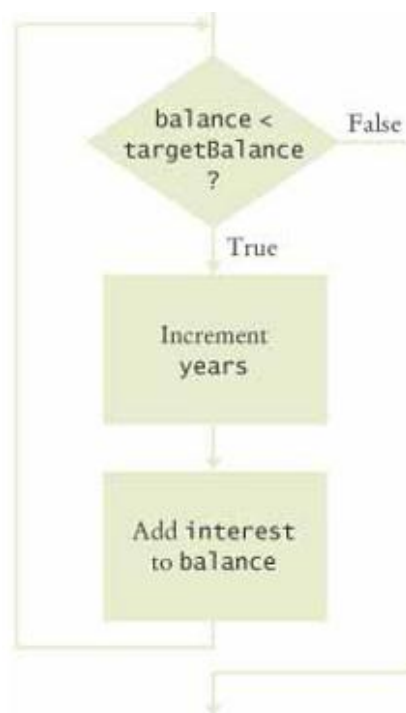
 The code will execute infinite times. Condition is a not equal to 2 means any value other than 2 is true. In start value of `counter` is 1 hence condition is true and flow goes inside loop body print value of `counter` and increment value of `counter` by 2 i.e. 1+2=3. Flow goes to condition again 3 not equal to 2 means true so flow once again goes inside loop print value of `counter` and increment value of counter by 2 i.e. 3+2=5. Similarly condition remains true and loop continues. Loop condition will remain true infinite times because addition of 2 in odd numbers always gives odd number; whereas; condition is counter not equal to 2 and 2 is an even number. This is example of logical error. Until you run the code, you can't know what you did by mistake.

There are two types of loops. One is counter loop, other is conditional loop. In this lecture we will discuss counter loop only.

Counter Loop:

A counter loop runs a fixed number of times. Like 10, 100, 500 or even more. A counter loop has typically 3 elements: **Initialization**, **Condition** & **Increment**. See following pseudo code to print first ten positive numbers:

```
counter = 1
while counter <= 10
     print counter
     counter = counter + 1
```

Here first element is "counter = 1". Second element is "counter <= 10" and last element is "counter = counter +1". Changing these 3 elements can affect number of times loop execute. Like in above code if we replace 10 with 50 in condition loop will execute 10 times. If we replace counter = counter + 2, loop will execute 5 times even counter condition is less than equal 10. Similarly initialization also effect number of times loop execute. Here counter is called loop controlling variable, because changing counter anywhere can change the number of time loop runs. Students must understand that loop controlling variable is required to control number of times loop executes. It is not necessary that same is also use in print statement. Consider following code:

```
counter = 1
while counter <= 10
     print "Forgive me my Lord"
     counter = counter + 1
```

In this code loop will execute same number of times but print "Forgive me my Lord" ten times. See another code:

```
counter1 = 1, counter2=10
while counter1 <= 10
     print counter2
     counter2 = counter2 + 10
     counter1 = counter1 + 1
```

Once again in this code loop will execute ten times but print 10, 20, 30, … , 100. There are two reasons: first we are printing counter2 and second is counter2 is incremented by 10 not by 1. Lastly see another example:

```
counter1 = 1, counter2=-1
while counter1 <= 50
     print counter2
     counter2 = counter2 - 1
     counter1 = counter1 + 1
```

This code will print first -1, -3, -5, …, -99 i.e. first 50 negative odd numbers; whereas counter1 goes from 1 to 50. Hence a loop controlling variable is a direct link with number of times a loop runs; whereas output is not directly link with controlling varaiable.

Syntax

We can create a java code having loop using while statement. Syntax of while is same as **if** statement but working is different. **while** in small letter is a keyword of Java language. while is followed by condition enclosed in parenthesis. Lastly there may be one statement with or without curly braces or block of statements with curly braces. See example:

```java
int i=1;
while (i<=5){
      System.out.println ("My Lord save me from hell fire.");
      i=i+1;
}
```

Output:

My Lord save me from hell fire.
My Lord save me from hell fire.
My Lord save me from hell fire.
My Lord save me from hell fire.
My Lord save me from hell fire.

See another program in java to print capital alphabets:

```java
char ch='A';
while (ch<='Z'){
      System.out.print (ch+" ");
      ch=(char)(ch+1);
}
```

Output:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Above program may seems strange to students but it is very simple. Here we have initialized ch used as controlling variable by 'A'. Condition is ch<='Z' i.e. continue till ch is less or equal to Z. Lastly ch is incremented by 1. By default the result of statement ch+1 is integer but we want to store it in ch which has char data type. Therefore type casting is required.

To get more understanding student must try this program with integer type variable as ASCII code of A is 65 and condition can be less than equal 90, the ASCII code of Z. Use type casting in print statement i.e. **"System.out.print ((char)count) "** .

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