وہی آسمانوں اور زمین کا پیدا کرنے والا ہے۔ جب کوئی کام کرنا چاہتا ہے تو اس کو ارشاد فرما دیتا ہے کہ ہو جا تو وہ ہو جاتا ہے۔ البقرۃ آیت ۱۱۷

Anyone who doesn't take truth seriously in small matters cannot be trusted in large ones either. Albert Einstein

Lecture 07

Decision Making using If Statement(s) & Programming Practice

In this lesson we will learn how to do decision making using if statements and later we will do programming practice.

Decision Making

Computer programs have to make decisions often and execute selective statements depending on the different inputs given at run time. For example finding grades of students comparing marks or percentage, checking balance before giving amount at ATM, giving discount if amount exceeds to mentioned amount in discount policy and sending marks comparing students roll no.

We will discuss about syntax part before that we should understand the philosophy of selection. Basically selection means to decide whether or not to execute any statement or block (set of statement) at run time. Essentially this means such programs have some statement (logic or relational) results in true/ false. A statement or block written after statement executes if condition has result true and otherwise skipped. See a pseudo code for more understanding:

```
-input n
- if n≠0
-    print "Reciprocal is:"
-    print 1.0/n
```

This is a small program to print reciprocal but reciprocal of 0 is undefined, ultimately result in crash of program. Therefore in this program we have a check that last 2 lines will executes only if n is not equal to 0, in order to manage crash of program. Below is the similar code with a different angle:

```
-input n
- if n equals 0
-    print "Reciprocal of 0 is undefined"
-    return
- print "Reciprocal is:"
- print 1.0/n
```

In previous program output comes if n is not equal to 0 but otherwise no output and user may not have any idea why output is not produced. In first program there are two statements at the end that may execute or otherwise program terminate. In second program we have written two statements which executes if n is 0 and program will terminate before going to last two statements. However, if n is non-zero the two statements after if statement will be skipped and last two statements will execute.

We may have another flavor of selection where we have two set of statement(s) and we select one of them based on the condition. We write one set of statements after if condition and write else phrase where second set of statements comes after else phrase. In this case only one set of statements will execute whatever condition is and one of the statements will definitely execute whatever input or condition is because if condition results in true the first set will execute otherwise second. Consider another code:

-input n - if n equals 0	-input n - if n not equals 0

- print "Reciprocal not defined" - else - print "Reciprocal is:" - print 1.0/n	- print "Reciprocal is:" - print 1.0/n - else - print "Reciprocal not defined"

Here two similar codes having same working only difference of writing shown on left and right side of table. In these codes either statements after if will execute or statements after else statement will executes. A last flavor in this lecture will be if-else chaining or if-else ladder. This type is pretty useful when we have to do more selection among more than two possibilities. For example we can think of finding student grades based on marks or percentage, another use is to calculate tax for different slab rates. In this case we may write individual if statements with composite condition. However, writing if-else chaining not only saves composite conditions also keep code simple and manageable. Just consider another pseudo code related to tax calculation:

- input s - if s<=100000 - print "No Tax" - if s<=200000 and s>100000 - print "Tax:", s*0.1 - if s<=300000 and s>200000 - print "Tax:", s*0.15 - if s>300000 - print "Tax:", s*0.2	- input s - if s<=100000 - print "No Tax" - else if s<=200000 - print "Tax:", s*0.1 - else if s<=300000 - print "Tax:", s*0.15 - else - print "Tax:", s*0.2

The codes on left and right has same functionality. In left side code we have to write composite condition because every condition is independent of other; whereas; in right hand side code we have not written a single composite condition also we have not written any condition in last else because if-else makes a combined functionality. In If-Else whenever a condition results in True next conditions are automatically skipped, therefore, not only save time but make conditions simpler.

If Syntax

We will discuss syntax in parts, starting from single if statement. We have two possibilities there may be single statement or set of statement called block. For a syntax possibility:

```
if (condition)
      statement
```

Syntax is **if** keyword in small letters followed by opening parenthesis followed by condition followed by closing parenthesis. A single statement or set of statements can follow if statement. If there is single statement then curly braces are optional but in case of multiple statements curly braces are mandatory. All of them given in following table:

if (total>10000)
      total = total – 1000;
//A discount of 1000 if total exceeds 10000
	if (total>10000){
	      discount = total * 0.1;
	      total = total – discount;
	}
	//A discount of 10% if total exceeds 10000

if (total>10000){
      total = total – 1000;
}	

In case of else part else keyword comes immediately after end of statement under if or end of curly braces. Similarly statement or block follow else statement. See example

```
if (total>10000)
      total = total - total * 0.10;
else
      total = total - total * 0.05;
```

//A 5% discount to all sales <= 10000 and 10% above 100000

```
if (total>10000){
      total = total - total * 0.10;
      ? ("10% discount");
}
else{
      total = total - total * 0.05;
}
```

Note: It is very important that if multiple statements written after if without curly braces else statement will show error "Else without if", therefore student should care to put curly braces whenever there are multiple statements occur.

The last syntax part is if-else chain. If-else chain in not much different from if-else except in second if statement **else if** is written with space between else and if and lastly else may come at the end of if–else chain. See example:

```
int marks=scan.nextInt();//input
if (marks < 50)
      gpa=0.0;
else if (marks < 55)
      gpa=1.0;
else if (marks < 58)
      gpa=1.7;
else if (marks < 61)
      gpa=2.0;
else if (marks < 65)
      gpa=2.3;
else if (marks < 70)
      gpa=2.7;
else if (marks < 75)
      gpa=3.0;
else if (marks < 80)
      gpa=3.3;
else if (marks < 85)
      gpa=3.7;
else
      gpa=4.0;
```

See PUCIT Grading System at http://www.pucit.edu.pk/index.php/academics

```
int marks=scan.nextInt();//input
if (marks < 50){
      gpa=0.0
      grade="F";
}else if (marks < 55){
      gpa=1.0;
      grade="D";
}else if (marks < 58) {
      gpa=1.7;
      grade="C-";
}else if (marks < 61) {
      gpa=2.0;
      grade="C";
}else if (marks < 65) {
      gpa=2.3;
      grade="C+";
}else if (marks < 70) {
      gpa=2.7;
      grade="B-";
}else if (marks < 75) {
      gpa=3.0;
      grade="B";
}else if (marks < 80) {
      gpa=3.3;
      grade="B+";
}else if (marks < 85) {
      gpa=3.7;
      grade="A-";
}else{
      gpa=4.0;
      grade="A";
}
```

See PUCIT Grading System at http://www.pucit.edu.pk/index.php/academics

In left hand code only gpa is calculated using single statement after each if; whereas; in right hand side code both gpa and grades are calculated therefore block was required.

Note: else statement is written after closing brace of previous if just to save space and is a valid syntax but it can be written in next line after closing brace as shown before.

More Programming Examples

We are assuming students are not familiar with basic programs and can write class name and main method, therefore, we are writing codes that can be placed inside main method.

1. Input 3 numbers and print which of them is largest:

```
int x, y, z;
Scanner in=new Scanner(System.in);
? ("X:");
x = in.nextInt();
? ("Y:");
y = in.nextInt();
? ("Z:");
z = in.nextInt();
if (x>y && x>z)
      ? (x +" is largest");
else if (y>z && y>x)
      ? (y +" is largest");
else
      ? (z +" is largest");
```

Students should replace first 3 ? by System.out.print and last 3 ? by System.out.println. In following examples we will assume that input is taken so we will give only if part.

2. Input 2 numbers and write them in order:

```
if (x>y)// Input already taken
      ? (x +", "+y);
else
      ? (y +", "+x);
```

3. Input 2 Words and write them in order:
   Before starting code student must understand comparison of strings, strings cannot be subtracted from each other, therefore, we can’t compare them using <, > symbols. Strings are compared character by character starting from first character. If first characters match then second characters are compared and it goes on. Whenever there is a mismatch of character comparison immediately stops. There are certain possibilities we discuss one by one:
   - if both strings have same number of characters and all characters match, means both strings are equal like `"Pakistan", "Pakistan"`
   - if one of the string has lesser length but characters are matching then string with less characters is smaller like `"Paki", "Pakistan"`. Here Paki is smaller
   - if characters mismatch irrespective of size of string, we can see ASCII codes of both character, the character with smaller codes will made respective string smaller even if it has more count of characters

   There are 3 possibilities in every comparison <, > and equal. We use ***compareTo*** method to compare two strings. Method return 0, value greater than 0 and value less than 0 depending on strings in comparison. `"abc".compareTo("abd")` returns negative number. `"abc".compareTo("ab")` returns positive number. `"abc".compareTo("abc")` returns zero. Hence we can compare to value ==0, >0 and <0. See example:

4. Input 2 words using next method of Scanner class and write them in order:

```
//Say s1 and s2 are two Strings
if (s1.compareTo(s2)>0)// Input already taken
      ? (s1 +", "+s2);
else
      ? (s2 +", "+s1);
```

5. Input 2 lines using nextLine method of Scanner class and print which of them has more characters:

```
if (s1.length()>s2.length())//
      ? ("First sentence has more characters");
else
      ? ("Second sentence has more characters");
```

See Next Lab for further practice, coming soon….