

اور نماز ادا کرتے رہو اور زکوٰۃ دیتے رہو۔ اور جو بھلائی اپنے لیے آگے بھیج رکھو گے، اس کو خدا کے ہاں پالو گے۔ کچھ شک

نہیں کہ خدا تمہارے سب کاموں کو دیکھ رہا ہے۔ البقرة آیت ۱۱۰

Lecture 04

Print and Variables

In this lesson we will learn some exceptional cases of print statement and use of variables in programs. Let's start with print statements Before this we have said that whatever you will write in double quotes is called String and same is printed as it is. Like:

```
print ("abc 123 +-* 3.6")    abc 123 +-* 3.6 (output)
```

So above print statement has output whatever written inside double quotes, but there are some exceptions. That is if we give \ backslash in double quotes it will not be printed rather it may give some special behavior depending on what comes after backslash. Note down that you can give limited characters after backslash; the list is given in table below. Each of the sequence in the list has different effect, short explanation.

Escape Sequence	Meaning	Explanation
\t	tab	cursor moves to next tab position, for further printing
\n	new line	cursor moves to start of next line, for further printing
\r	carriage return	cursor moves to start of same line and further printing may overwrite contents already printed on same line
\b	backspace	erase 1 character towards left and continue printing towards right
\'	single quote	print ' single quote
\"	double quote	print " double quotes
\\	backslash	print \ backslash

Let's consider program and its output to get more understanding of escape sequence.

<pre>class PrintUsingEscapeSequence{ public static void main(String []args){ System.out.println("1.\n2.\n3."); System.out.println("1234\r56"); System.out.println("Hamid\t83"); System.out.println("Khawar\t73"); System.out.println("Nida\t78"); System.out.println("Junaid\t65"); System.out.println("Sum==\b23"); System.out.println("He has given \"very\" +\" good\" comments."); System.out.println("Let\'s walk!."); System.out.println("He has given Rs. "+ "1000 \\\-"); } }</pre>	<p>Output:</p> <pre>1. 2. 3. 5634 Hamid 83 Khawar 73 Nida 78 Junaid 65 Sum=23 He has given "very good" comments. Let's walk!. He has given Rs.1000\-</pre>
---	--

We will discuss print statements one by one:

- i. `\n` is taking next printing to next line, so you can see each number is coming in next line
- ii. after printing 1234 `\r` takes cursor back to start of same line and start printing 56, therefore you can see only 5634 because 56 is overwritten over 12
- iii. `\t` in next statements taking next output to next tab position in result you can see numbers in front of names are aligned
- iv. Here String Sum== is printed but `\b` erase last character that is = and print 23, therefore you can see only one equal character instead of two characters
- v. In next statement you can see very good is enclosed using double quotes, where in code `\"` is written to print "
- vi. Similarly in next line you can see single quote
- vii. Lastly `\\` two backslashes are written to print one `\` backslash

Note: Lastly student should understand and practice by writing output manually than cross check through program output. Students should write the same output that is produced by the program on console. That is they should not write parenthesis or double quotes etc.

Variables & Variable Types

In this portion we will learn what are variables, where to use them and how to use them and finally some types of variables in Java. Just try to recall memory button you were using in your calculators in your high school or may be before. The value you need to use later you store in calculator memory by pressing M+ button and later you can find the same by pressing MR button of calculator. There are many uses of memory, for example if you want to calculate $\frac{\sin(x+y)}{\cos(x-y)}$, you can first calculate **x-y**, than take **cosine** of same and **save** it into memory, later calculate **x+y** and take **sine**, now **divide** the result by value already saved in memory. Another case is if you have a common sub-expression in multiple expressions you may save common sub-expression into memory and later instead of typing you can use it by pressing memory recall button.

Computer has two types of memory primary/ volatile/ internal and secondary/ non-volatile/ external. Internal memory is expensive, fast and smaller; whereas; external memory is relatively cheaper, very slow, and relatively very large. For example now a days computer comes with 2-4 GB internal memory called (RAM) having cost around 1500 to 3000 rupees; whereas; computers have external memory called (Harddisk) around 500 GB having cost around 5000 rupees. Similarly speed of RAM is around 1000 MHz (million per second); whereas speed of external memory is around 7200 RPM (rounds per minute). Computer internal memory is temporary/ volatile can be used till computer is in working state, will be lost when computer turned off or restart; whereas; external memory is permanent. Therefore, RAM is used for storing during program execution; whereas; external memory is used to store data which can be used later by the same program or may be by some other program. An important thing is that both of these types of memories can be rewriteable therefore if you rewrite at the same location it will erase previous value and new value will be stored.

Memory location has addresses like 1000 or 1010 but we give names to those locations for our convenience. There are two reasons for that one is that exact memory address is given to programs at run time normally because now a days a number of programs can run in a computer simultaneously. The second reason is it is difficult to remember to use memory locations whereas if we give some reasonable names to memory location it is very much convenient to remember. Some variable names can be sum, width, height, weight, count, average, amount, percentage, city, capital, designation, salary etc. Now just consider why to use memory for example if you have to write a program to print your beautiful name with stars surrounding like:

***** Your Name Here *****

you need to write just print statement that is sufficient. Similarly to print sum of two numbers you may again write print statement like `print (2+3)` and also you may write `print ((2+3)/2.0)` to print average of two numbers but just suppose if you want to write both sum and average, you have to write addition expression two times like `print (2+3)` and `print ((2+3)/2.0)`. It will be more convenient if you use variable to store addition expression and use it to calculate average and also to print sum as well like:

```
sum = 2+3
average = sum / 2.0
print (sum)
print (average)
```

This is small example the level of complication will be increased as we have to write more statements and lastly it is impossible to work without variables if your program requires user input. Just consider example:

```
input x
input y
sum = x + y
average = sum / 2.0
print (sum)
print (average)
```

It is very important to remember that we don't need separate variable for every sub-expression, rather we may use same variable again if previous value is no more required. For example to

calculate $\frac{4x^3}{y}$:

```
val = x*x*x
val = 4 * val
print (val/y)
```

...

Variable/ Data Types

Each language gives different types to store data according to type. The first purpose of variable types is to be store and retrieve data according to type. Just for example if we store 5 in "**int**" type variable it will just compute the binary that is "101" and store into memory "0...0101" in memory; whereas if we say store 5 in "**double**" type variable the storage will be entirely different. Similarly if we store A in character type variable, it will take *ascii* of A that is 65 then convert it into binary that is "0100001" and store in memory. Secondly purpose of type is to perform operations according to data type like if division is performed between two **int** type variables the result is different; whereas if one of them or both are **double** result will be different. Similarly if + operator is written between int type variables or double type variables or a mix of them it will perform addition; whereas; if one or both of them are String it will perform concatenation. Java has many data types but right now we will discuss only three data types:

Type	Size	Purpose	Range
int	4	to store integers	-2,147,483,648 to 2,147,483,647
double	8	to store real numbers	4.9 E-324 to 1.79 E308
boolean	1	to store true/ false	true/ false

Next we will discuss taking input in Java than we will see a program to see all above in working. Two lines to be added in every program to take input from user. The very first line of your program should be:

```
import java.util.*;
```

This line will come before class name. The second line may come inside main method:

```
Scanner input=new Scanner(System.in);
```

The code of program TestVariables is given:

```
import java.util.*;

class TestVariables{
    public static void main(String []args){
        Scanner input = new Scanner (System.in);
        int x1, x2, sum;
        double average;
        boolean res1, res2;
        System.out.print ("Enter X1:");
        x1 = input.nextInt();
        System.out.print ("Enter X2:");
        x2 = input.nextInt();
        sum=x1+x2;
        average=sum/2.0;
        res1=x1>average;
        res2=x2>average;
        System.out.println (x1+" "+x2+" "+sum);
        System.out.println ("Average:"+average);
        System.out.println ("Result1:"+res1);
        System.out.println ("Result2:"+res2);
    }
}
```