Chapter 5:

Advanced Programming Concepts

Informatics Practices

Class XII (CBSE Board)

Revised as per CBSE Curriculum 2015

"Open Teaching-Learning Material"

Visit www.ip4you.blogspot.com for more….

Authored By:- Rajesh Kumar Mishra, PGT (Comp.Sc.)
Kendriya Vidyalaya Upper Camp, Dehradun (Uttarakhand)
e-mail : rkmalld@gmail.com

Classes & Objects in Java

- □ In general, Class represents a collection (group) of similar types of entities e.g. Human, Bird, Car etc. Whereas an Object is a member of a Class like Student and Teacher (since belongs to Human).
- □ In OOP terminology, a Class encapsulates (binds) data and methods members in a single unit and an Object is an instance of a class that holds actual data in memory.
- □ JAVA offers a massive collection of ready-to-use Class libraries containing reusable components/methods in the form of Packages, that can be used in an Application/ program to perform predefined task.
- □ Some commonly used Java libraries are Math Library, String Library, Utility Library and I/O Library etc. available in various packages.

JAVA Libraries

- To save our time and programming efforts, Java offers a collection of ready-to-use Libraries of Classes (called Packages) which can be used directly in the Applications.
- The Packages containing classes, are used by importing them in an Application/Program as per our need. You can use import statement at the top of the program to include the Java libraries, as-

```
        import java.io.*;        // to use classes for I/O methods
```

- The **java.lang** package contains general purpose classes for Mathematical and String operations. It is default package and imported automatically in a program i.e. imported without writing import statement.

Math Class & its commonly used methods

- Java provides **Math Class** (Library), which available in **java.lang** package. Math class contains built-in methods for performing basic numeric operations such as Exponential, Rounding, Square root and Trigonometric functions.

- In order to use functions/methods of math library, you need to invoke function using Math keywords before the function.

e.g. x=Math.sqrt(25);

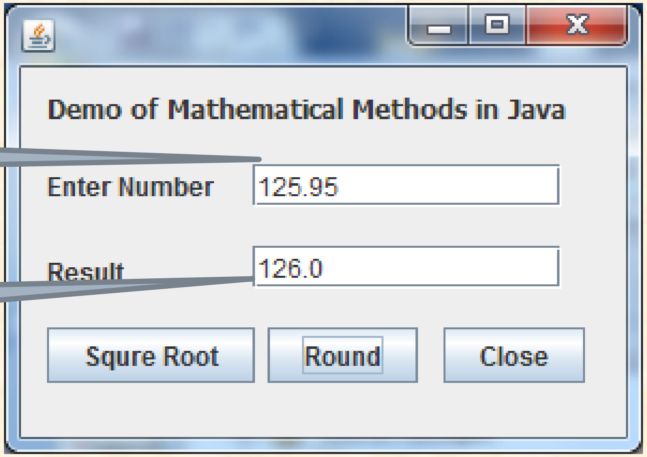
Method Prototype	**Description**
pow(num1,num2)	**It computes num1 ^{num2} , where num1 and num2 are numbers.** **e.g. syste.out.print(“”+Math.pow(2,3);**
round(num1)	**It rounds off a given number to its nearest integer. It can take float/double as argument.** **e.g.** **system.out.print(“”+Math.round(1.5)); 2** **system.out.print(“”+Math.round(-1.5)); -1**

Demonstration of Math Class & its methods

Consider the application as given below-

TextField1

TextField2



Demo of Mathematical Methods in Java

Enter Number 125.95

Result 126.0

```
// TODO Code for Squre Root Button//
        double num1,num2;
        num1=Double.parseDouble(jTextField1.getText());
        num2=Math.sqrt(num1);
        jTextField2.setText(""+num2);
```

```
// TODO Code for Squre Root Button//
        double num1; float num2=0;
        num1=Double.parseDouble(jTextField1.getText());
        num2=Math.round(num1);
        jTextField2.setText(""+num2);
```

String Class :

- □ String Class includes methods for converting strings into lower case or upper case, extracting substrings, joining two strings together, calculating the length of string and removing leading and trailing spaces etc.
- □ In Java, String is group of characters in a memory like a queue and each character has assigned a position or index starting from 0.
- □ For example a string “**Hello Java**” is represented as -

H	e	l	l	o		J	a	v	a
0	1	2	3	4	5	6	7	8	9

Index

Length of string is 10

Note : Space is also counted as a character.

Commonly used methods of String Class :

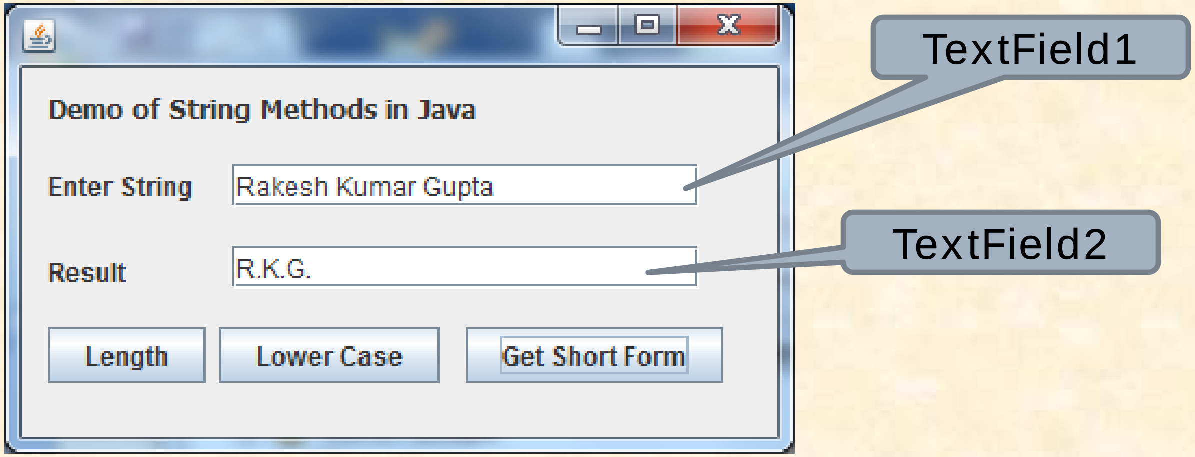
Method	Description
.equals(str)	Compare this (current) string to given string and returns true if both are true otherwise false. e.g. boolean test=str1.equals(str2);
.length()	Returns the length of this string as a number. e.g. int x=str1.length();
.substring(num1[,num2])	Returns a substring from num1 to num2 position (character at num2 position is excluded). If num2 is missing then string from Num1 to End position is given. e.q. String st="abcdefg".substring(2,5); It will give "cde" not "cdef".
.concat(str)	Return a string after appending str into this string. e.g. String st= "Hello".concat("Java"); It will gives "HelloJava"
.toString()	Converts the given object as String. e.g. String st= Integer.toString(5);

Commonly used methods of String Class :

Method	Description
.toLowerCase(str)	Coverts all the characters of this string into lowercase. e.g. String str2=str1.toLowerCase();
.toUpperCase(str)	Coverts all the characters of this string into Upper case.
.trim(str)	Returns a string after removing leading and trailing spaces. e.g. String str2=str1.trim();
.charAt(num)	Returns the character at given position in this string. e.g. char ch=str1.charAt(3);
.indexOf (chr)	Returns a number as position of given character into this string. e.g. int x= str1.indexOf(‘A’);
.replace(char1,char2)	Returns a new string after replacing all occurrences of char1 by char2. e.g. String str2= str1.replace(‘a’);

Demonstration of String Class & its methods

Consider the application as given below-



Demo of String Methods in Java

Enter String: Rakesh Kumar Gupta

Result: R.K.G.

```
// TODO Code for Length Button//
        String str= jTextField1.getText();
        int l=str.length();
        jTextField2.setText(""+l);
```

```
// TODO Code for Lower Case Button//
        String str1= jTextField1.getText();
        String str2 =str1.toLowerCase();
        jTextField2.setText(str2);
```

Demonstration of String Class & its methods

```
// TODO Code for Get Short Form Button//
        String str1,str2,sn;
        str1= jTextField1.getText();
        //get first character and add . //
        sn=str1.substring(0,1);
        sn=sn.concat(".");
        //make a loop upto last position//
        for (int i=1;i< str1.length();i++)
        { /* test whether a character at ‘i’ position is space
           and next character is not space */
          if(str1.charAt(i)==' ' && str1.charAt(i+1)!=' ')
             {
               /*extract character just after space and add to sn
               variable alongwith ‘.’ */
              str2 = str1.substring(i+1,i+2);
              sn = sn.concat(str2);
              sn = sn.concat(".");
             }                  // end of condition
        }                       // end of loop
        jTextField2.setText(sn.toUpperCase());
```