CHAPTER 6: COMMONLY USED LIBRARIES

Libraries: It is a set of ready-made software routines that can be reused in new programs. Libraries are made available in a program using import statement e.g. import java.io.*;

String Library: It is a part of java language library java.lang, which is by default imported to the program. **There are 3 classes to work with characters:**

(i) **Character Class** whose instances can hold single character data.

(ii) **String Class** whose instances can hold unchanging string. Every time we change, a new object is created automatically.

String s = "Excellent" ; or String s = new String("Excellent");

(iii) **StringBuffer Class** whose instances can hold mutable strings.

StringBuffer sb = new StringBuffer(); or StringBuffer sb = new StringBuffer("First"); or
StringBuffer sb = new StringBuffer(10); // to hold 10 characters
String s = "crack", r = "rack";

Method Prototype	Description
String toString()	r = s.toString(); Returns the string itself.
String concat(String)	Concats two strings. s = s.concat(r); -> crackrack
str1 + str2	Concatenation operator. s = s + r ;
int length()	n = s.length() -> 5
String toLower Case()	r = s.toLowerCase();
String toUpperCase()	r = s.toUpperCase();
String trim()	r = s.trim(); Removes white spaces from both ends.
String substring(int beginindex, int endindex)	s.substring(1,3); =>> ra. Inclusive of start index and exclusive of end index. Index starts at 0.
boolean equals(str)	b = s.equals(r);
boolean equalsIgnoreCase(str)	b = s.equalsIgnoreCase(r);
String valueOf(all types)	r = s.valueOf(r); Returns string representation of passed argument i.e. 30 is returned as "30".
valueOf()	int i = Integer.valueOf(k); Returns Integer representation of passed argument i.e. "30" is returned as 30. double x = Double.valueOf(k); //x becomes 30.0

String s = "art", r = "science";
StringBuffer sb = new Stringbuffer(s);

Additional stringBuffer Methods:

Method Prototype	**Description**
append(x)	Adds x char at the end. sb.append(r); =>> artscience.
reverse()	sb.reverse(); =>> ecneicstra

Math Functions: It is found in Math library and used as Math.sqrt(a*a + b*c)

Function	Action	Function	Action	Function	Action
pow(x,y)	X^y	**ceil(x)**	Round up	**floor(x)**	Round down
sqrt(x)	Root	**abs(a)**	Absolute	**max(a,b), min(a,b)**	Max, min
round(x): Rounds off a number to its nearest integer. It argument is double then it returns long and for float it returns int. If the argument is NaN, then the result is zero. Round(-4.5) is -4. Round(4.5) is 5. (NaN means Not a Number).					

Sample Questions:

1. What will be display in a jTextField1 after executing the following code?

```
int m= 16;      m=m+1;
if (m<15)      jTextField. setText(Integer.toString(m));      else
jTextField1.setText(Integer.toString(m+15));
```

2. What does round() return if a negative float value is passed to it?
3. Write code to display IP 12 CBSE in a dialog box (JOptionPane).
4. What will be the output of the following code: (i) Math.round(1.5) (ii) “Welcome”.toUpperCase().
5. What will be the value X1 after the execution of the following code ?

```
String X1= "Graduate" ,X2="Post" ;    X1=X2. concat (X1) ;
```