STRING CLASS AND PRINTING

String object:

- String is a built-in class available in java.
- A string is a collection of characters.
- Simplest method for creating a string is String str1="Java program";
- Variable of type object are called as reference.
- String object is referred to as a literal.
- References are used for holding or pointing objects.
- String literal is taken in " ".
- Keyword new is taken to create an object in heap.
- Constructors are methods or functions used for creating string object.
- There are three types of constructors:
  - String(char [ ]).

    Array of characters is taken and is converted into string.
  - String(byte [ ]).

    Array of bytes is taken and is converted into a string.
  - String(string).

    A string literal is taken in this method.

    New string is created using this method.
- Java maintains a pool of string constants.
- Whenever new is applied the object is created in heap memory and the memory occupied by the object literal is in pool.

- When same literal is used java does not create another object.
- When new is applied the object in pool may or may not be created.
- String objects are immutable/ they cannot be modified.

Methods of string class:

- The methods creates a new string before giving the results.
- The new object is then created in heap memory.

#1.

- str.length(): it returns the length of the string mentioned.
- str.toLowerCase(): it converts the given characters of string into lower case.
- str.toUpperCase(): it converts the given characters of string into upper case.
- str.trim(): it is used to remove the leading and tailing spaces from the array if there are any.
- str.substring(int begin): it returns a new string by giving the part of a string from where the index is given.
- str.substring(int begin, int end): it works same as the above but the starting and ending index both can be given in the substring.
- str.replace(char old, char new): it replaces the old character with the new character.

#2.

- str.startsWith(string s): to find the particular starting word of a the string/to find the initials.
- Str.endsWith(string s): to find particular ending word of the string.
- str.charAt(int index): to find the particular character present on the index given.
- str.indexOf(String s): to find the index of the given character.
- str.lastIndexOf(string s): to find the index of the given character from the end of the array.

#3.

- str.equals(string s): to check whether the contents of two strings are equal or not.
- str.equalsIgnoreCase(string s): to check whether the contents of two strings are equal or not but it does not depend upon the case of characters.
- str.compareTo(string s): to compare two strings according to the dictionary order(in accordance with the ASCII codes for cases of characters).
- str.valueOf(int i): to convert different types of data into string data type.

Regular Expressions:

They are used to define the symbols.

Matching symbols:

- these symbols are meant for single alphabets.
- '.' it means any letter or the symbol from the keyboard.i.e for single alphabet it is true.
- [abc]: range or set of characters/ the string is true if the alphabet is either a or b or c.
- [abc][vz]: range of multiple symbol/the string is true if first symbol is among a,b &c and second symbol is among v & z.
- [^abc]: the string is true if the symbol is other than a, b, & c.
- [a-z 1-7]: the string is true if the symbol is from the range a-z or 1-7.
- A|B: it is true for single alphabets fro either A or b.
- XZ: to check whether the string maybe either a single or multiple alphabets.

Meta characters:

- \d: it will be true if it is a digit among 0-9.
- \D: it will be true if it is any symbol other than digits.
- \s: it will be true if there is just a space.
- \S: it will be true if there are any symbols other than space.
- \w: it will be true if it is either alphabet or digits.
- \W: it will be true if it is any symbol other than alphabets or digits.

String matching with regular expressions:

- Quantifiers : these are used to define the number of symbols you want.
    - '*' it represents number of occurrences of any of the characters for zero or more times.
    - '+' it represents number of occurrences of any of the character for one or more times.
    - '?' it represents number of occurrences of any of the characters for zero or one time.
    - {X} it represents any of the characters for the size of X value given.
    - {X, Y} it represents any of the characters for the min and max size given.