```
package String_Examples;
import java.lang.Object.*;
public class Example1
{
    public static void main(String[] args)
    {
        String s1="Pune Institute of Computer Technology";
        String s2="Pune";

        //1) length(): To find out the no char in the given string
        System.out.println("String length="+s1.length());//Result:-37

        //2)concatenation:- concat()
        System.out.println("concatenated result:"+(s1+" "+s2));//PICT
Pune
        System.out.println(s1.concat(s2));//Result:-PICTPune

        //3)String to CharArray():-used to convert all the characters
        // of a string into a Character Array
        char ele[]=s1.toCharArray();
        System.out.println(ele[2]);//Result:-n
        for(char c:ele)
        {System.out.print(c);}//print every char
        System.out.println("");

        //4)String charAt():This method is used to retrieve a single
        // character from a given String.
        System.out.println(s1.charAt(3));//e
       //System.out.println(s1.charAt(50));//Result:-exception
       // System.out.println(s1.charAt(-1));Result:- exception

        //5)String compareTo(): method is used to compare two Strings.
        // The comparison is based on alphabetical order.
        System.out.println(s1.compareTo(s1));// result:-0
        System.out.println(s1.compareTo(s2));//result:-33

        //6)String contains():-method is used to determine whether a
        // substring is a part of the main String or not.
        // The return type is Boolean.
        System.out.println(s1.contains(s2));//Result:- true

        //7)String split():-a split() method is used to split or separate
        // the given String into multiple substrings separated by the
        // delimiters (“”, “ ”, \\, etc).
        String s1array[]=s1.split(" ");
        for(String s:s1array) {System.out.println(s);}
        //result:- display each word of the string separately


        //8)String indexOf():search particular char or substring in the
        // main string from specific index also
        //indexOf() is used to search for the first occurrence of the
character.
```

```
       //lastIndexOf() is used to search for the last occurrence of the
character.
        System.out.println(s1.indexOf('P',10));//Result:-1
        System.out.println(s1.indexOf('P'));//Result:0
        System.out.println(s1.indexOf('o'));//Result:15
        System.out.println(s1.lastIndexOf('o'));//Result:34

        //9)JAVA String toString():convert another object into string
object
        Integer num=Integer.valueOf(10);
        System.out.println(num);//result:10
        String s3=num.toString();
        System.out.println(s3);//result:string 10

        //10)String reverse():reverse method is not preset in the string
class
        //which is present in the StringBuffer or StringBuilder class
        StringBuffer s=new StringBuffer("Pune");
        System.out.println(s.reverse());//result:enuP

        //11)String replace()
        //replace() method is used to replace the character with the
        // new characters in a String.
        String s1replace=s1.replace('P','M');
        System.out.println(s1replace);//result: First P replace by M
        s1replace=s1.replace("Pune","Mumbai");
        System.out.println(s1replace);//result:Pune replace by Mumbai
        s1replace=s1.replaceAll("o","i");
        System.out.println(s1replace);//result:every o replace by i

        //12)String substring():Substring() method is used to return the
substring
        // of the main String by specifying the starting index and the
        // last index of the substring.
        System.out.println(s1.substring(0,4));

        //13)String toUpperCase():
        System.out.println(s1.toUpperCase());
        //14) String toLowerCase():
        System.out.println(s1.toLowerCase());

        //15)String equals(): compare the two string,they are equal or
not
        System.out.println(s1.equals(s2));//result:false
        System.out.println(s2.equalsIgnoreCase("pune"));//result:true

        //16)String repeat()
        System.out.println(s2.repeat(2));//result:PunePune

        //17)String hasCode():return the hashcode value of the string
        System.out.println(s2.hashCode());//result:2499228

        //18)String Class strip() method returns a string that provides
        // a string with all leading and trailing white spaces removed.
```

```
        // This method is similar to the String.trim() method.
        String s4="  Pune Maharashtra   ";
        System.out.println(s4.strip());
        System.out.println(s4.stripLeading());
        System.out.println(s4.stripTrailing());

        //19)String trim():remove the leading and trailing the white
spaces
        System.out.println(s4.trim());

        //20)How to remove the all white space from the given string
        System.out.println(s1.replaceAll("\s",""));
        //remove multiple white spaces
        String s5=" Pune   Maharashtra ";
        System.out.println(s5.replaceAll("\s+",""));
        //result:input string display without spaces
        // \s:represent the single space in the string
        // \s+:represent the multiple white spaces in the string
        //Another logic
        String s6="";
        for(int i=0;i<s1.length();i++)
        {
            char ch=s1.charAt(i);
            if(!Character.isWhitespace(ch))
            {s6+=ch;}
        }
        System.out.println(s6);

        //21)show ascii value of any index char
        System.out.println(s1.codePointAt(3));//result:101
        System.out.println(s1.codePointBefore(3));//result:110

       //22)replace particular char in uppercase or lowercase
        String s7="";
        for(int i=0;i<s1.length();i++)
        {
            char ch=s1.charAt(i);
            if(Character.isUpperCase(ch)){s7+=Character.toLowerCase(ch);}
            else{s7+=ch;}
        }
        System.out.println(s7);

        //23)replace the digit in the string with addition result of
digit+1
        String s8="Pu4ne";
        String s9="";
        for(int i=0;i<s8.length();i++)
        {
            char ch=s8.charAt(i);

if(Character.isDigit(ch)){s9+=Character.getNumericValue(ch)+1;}
            else s9+=ch;
        }
        System.out.println(s9);
```

```
        //24)occurrence of the char in the given string
        String s10="ABCD ABCD ABCD ABCD";
        char array[]=s10.toCharArray();
        for(int i=0;i<s10.length();i++)
        {
            int count=0;
            char ch=s10.charAt(i);
            if(!(i>s10.indexOf(ch)))
            {
                for(int j=0;j<array.length;j++)
                {
                        if(s10.charAt(i)==array[j])
                        {
                            count++;
                        }
                }
                if(s10.charAt(i)=='\s')
                {
                    System.out.println("the occurrence of white space
is="+count);
                }
                else
                    System.out.println("the occurrence of
"+s10.charAt(i)+" is="+count);

            }
        }
       /*result:
       the occurrence of A is=4
       the occurrence of B is=4
       the occurrence of C is=4
       the occurrence of D is=4
       the occurrence of white space is=3*/
    }
}
```