④ Method Chaining :-

Method chaining is the practice of calling different Methods is a single line instead of calling other methods with same object reference seperately.

Syntax :- obj.method1().method2().method3()

Eg ① :-

```
String name = "Sachin";
String data = name.toUpperCase();
int Count = data.length();
System.out.println(Count);
```

// Method chaining.

```
System.out.Print(name.toUpperCase().length());
```

Eg ②

```
StringBuffer sb = new StringBuffer("virat");
S.o.P(sb.append("kohli").insert("10, "anushka"));
```

Note :- Method chaing applied on object with same return type.

→ append and insert can be applied on StringBuffer object.

```
sb.append("Kohli").insert(10, "Anushka");
```

Method I Method II

→ Return type of Method I and object on this Method invoked should be Same.

Eg ②

→ append method is ~~StringBuffer~~ return type is StringBuffer so insert Method can be applied on this object.

Eg ③

```
StringBuffer sb = new StringBuffer("Virat");
sb.append("Kohli")          → ViratKohli
  .insert(10,"Anushka")     → ViratKohliAnushka
  .reverse()                → akhsunAilhoktariV
  .append("IND")            → akhsunAilhoktariVIND
  .insert(sb.length(), "RCB")
                            ↳ akhsunAilhoktariVINDRCB
  .reverse().delete(0,6);   → ViratKohliAnushka
S.o.P(sb);
```

BCRDNIViratKohliAnushka

output : ViratKohliAnushka

Example ③ :-

```
StringBuffer sb1 = new StringBuffer("dhoni");
sb1.length().append("csk");
```

Return type of length is int

append method is not applicable on integer type data.
it gives Error

output :- Cannot invoke append(String) on the Primitive type int.

Problems

① Copy one string to another String

```
String s1 = "ineuron";
String s2 = "";
for(int i=0; i<s1.length(); i++)
{
    s2 = s2 + s1.charAt(i);
}
System.out.println("first string is" + s1);
System.out.println("Copy of first string" + s2);
```

↳ one by one characters are copying of s1 in s2

↳ output : ineuron

② Lower Case to upperCase & upper Case to Lower Case

① i/p: ineuron o/p: INEURON
② i/p: INEURON o/p: ineuron.

logic :-

ASCII Value
a ⇒ 97
A ⇒ 97 - 32

```
char ch = 'a'; // 97
S.O.P(ch) ⟶ // output : a
ch = (char)(ch - 32);
S.O.P(ch); // output : A
```

(i) lower Case to upperCase

```
Scanner Scan = new Scanner(System.in);
S.o.p("Enter String");
String S1 = Scan.nextline();
String S2 = "  ";
for(int i=0; i<=S1.length(); i++);
{
    S2 = S2 + (char)(S1.charAt(i)-32);
}
S.o.P(S2);
```

↳ int to char Conversion.

input : ineuron
output: INEURON

(ii) upperCase to lower Case :-

```
String S1 = "INEURON";
String S2 = "  ";
for(int i=0; i<=S1.length(); i++){
    S2 = S2 + (char)(S1.charAt(i)+32);
}
S.o.P(S2); // output : ineuron
```

(3) To Convert Small to Capital and Vice Versa

```
String S1 = "iNeuRON";
String S2 = "  ";
for(int i=0; i<S1.length(); i++){
    if(S1.charAt>='a' && S1.charAt<='z')
    {
        S2 = S2 + (char)(S1.charAt(i)-32);
    }
    Else if(S1.charAt>='A' && S1.charAt<='Z')
    {
        S2 = S2 + (char)(S1.charAt(i)+32);
    }
}
S.o.P(S2); // output : INEURON
```

(4) Possible ways to reversing a string

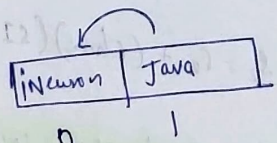
(i) iNeuron --> norueNi

```
String s1 = "iNeuron";
String s2 = "";
for(int i = s1.length()-1; i>=0; i--)
{
    s2 = s2 + s1.charAt(i);
}
s.o.p("original string" + s1);
s.o.p("After Reversing" + s2); // output: norueNi
```

(ii) iNeuron Java --> norueN avaJ

↳ it is done by using .Split Method.

↳ it Split two words and Create an array..



(iii) iNeuron java --> norueN avaJ

```
String s1 = "INeuron java";
String s2 = "";
String sarr[] = s1.Split(" ");
for(String elem : sarr)
{
    for(int i = elem.length()-1; i>=0; i--)
    {
        s2 = s2 + elem.charAt(i);
    }
    s2 = s2 + " ";
}
s.o.p(s2); // output: norueN avaJ
```

③ Ineuron java --> java Ineuron

```
String s1 = "Ineuron java";
String s2 = " ";
String sarr[] = s1.split(" ");
for(int i = sarr.length-1; i>=0; i--){
        s2 = s2 + sarr[i] + " ";
}
s.o.p(" ");
s.o.p(s2); // output : java Ineuron
```