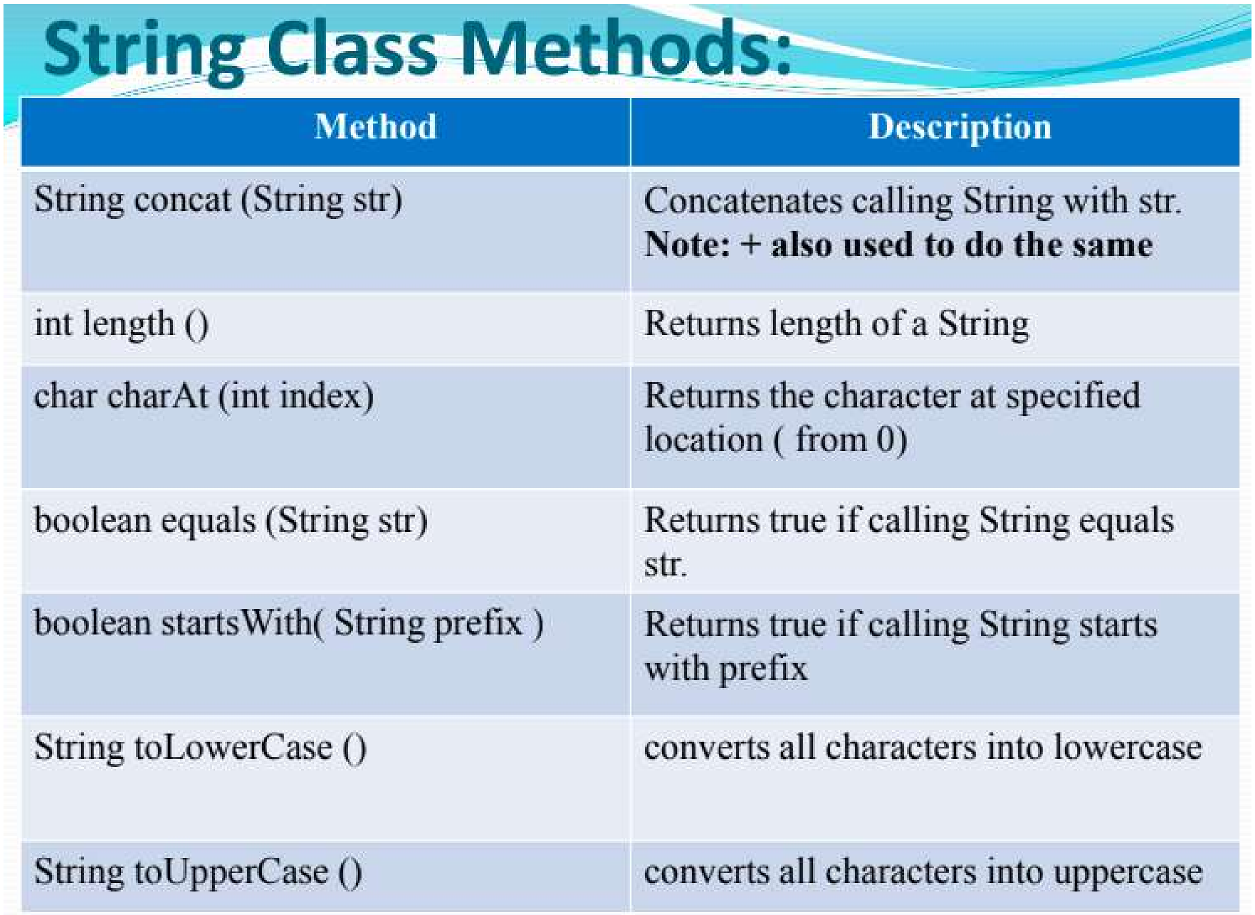
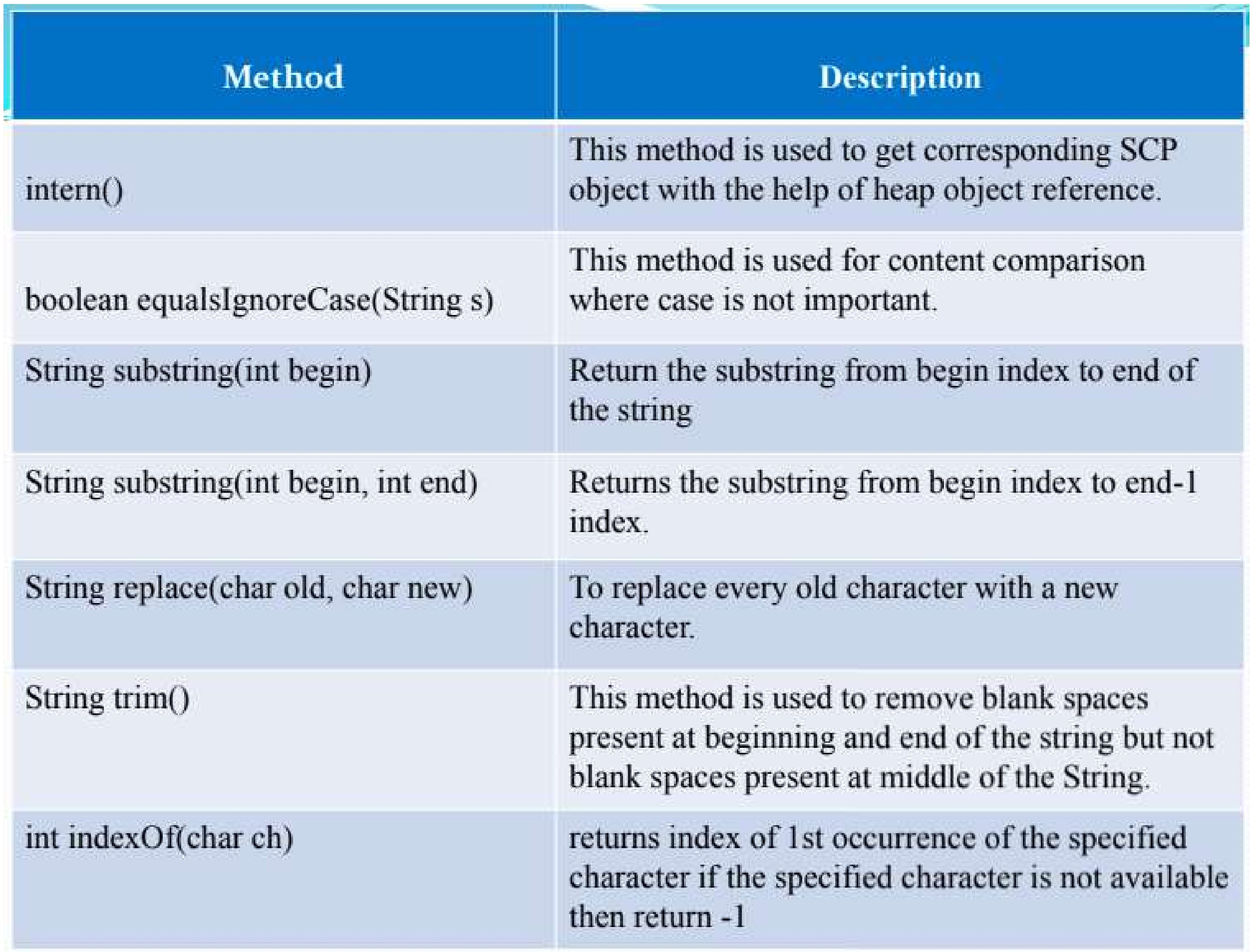


String Class Methods:

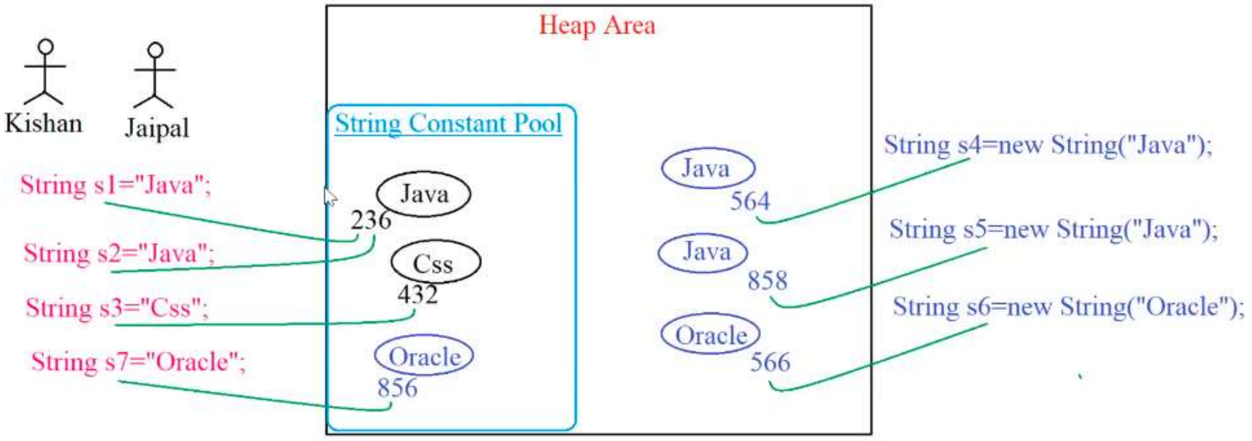
Method	Description
String concat (String str)	Concatenates calling String with str. **Note: + also used to do the same**
int length ()	Returns length of a String
char charAt (int index)	Returns the character at specified location (from 0)
boolean equals (String str)	Returns true if calling String equals str.
boolean startsWith(String prefix)	Returns true if calling String starts with prefix
String toLowerCase ()	converts all characters into lowercase
String toUpperCase ()	converts all characters into uppercase



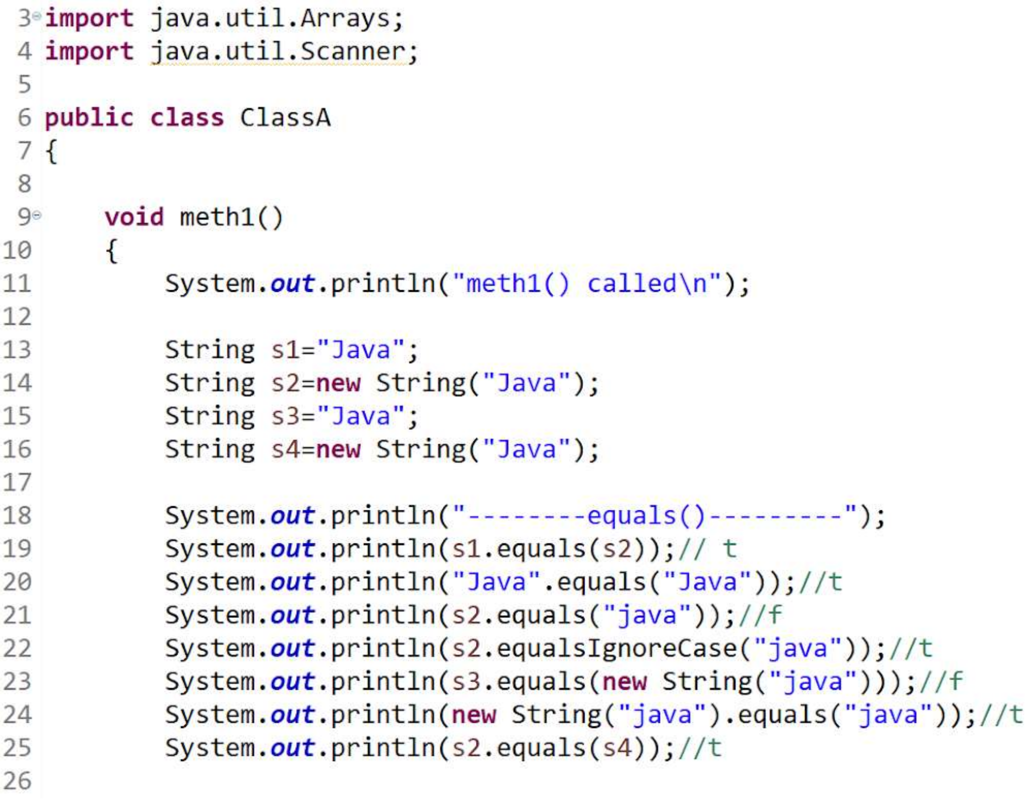
Method	Description
intern()	This method is used to get corresponding SCP object with the help of heap object reference.
boolean equalsIgnoreCase(String s)	This method is used for content comparison where case is not important.
String substring(int begin)	Return the substring from begin index to end of the string
String substring(int begin, int end)	Returns the substring from begin index to end-1 index.
String replace(char old, char new)	To replace every old character with a new character.
String trim()	This method is used to remove blank spaces present at beginning and end of the string but not blank spaces present at middle of the String.
int indexOf(char ch)	returns index of 1st occurrence of the specified character if the specified character is not available then return -1

String s="Java"; ⟹ Case 1

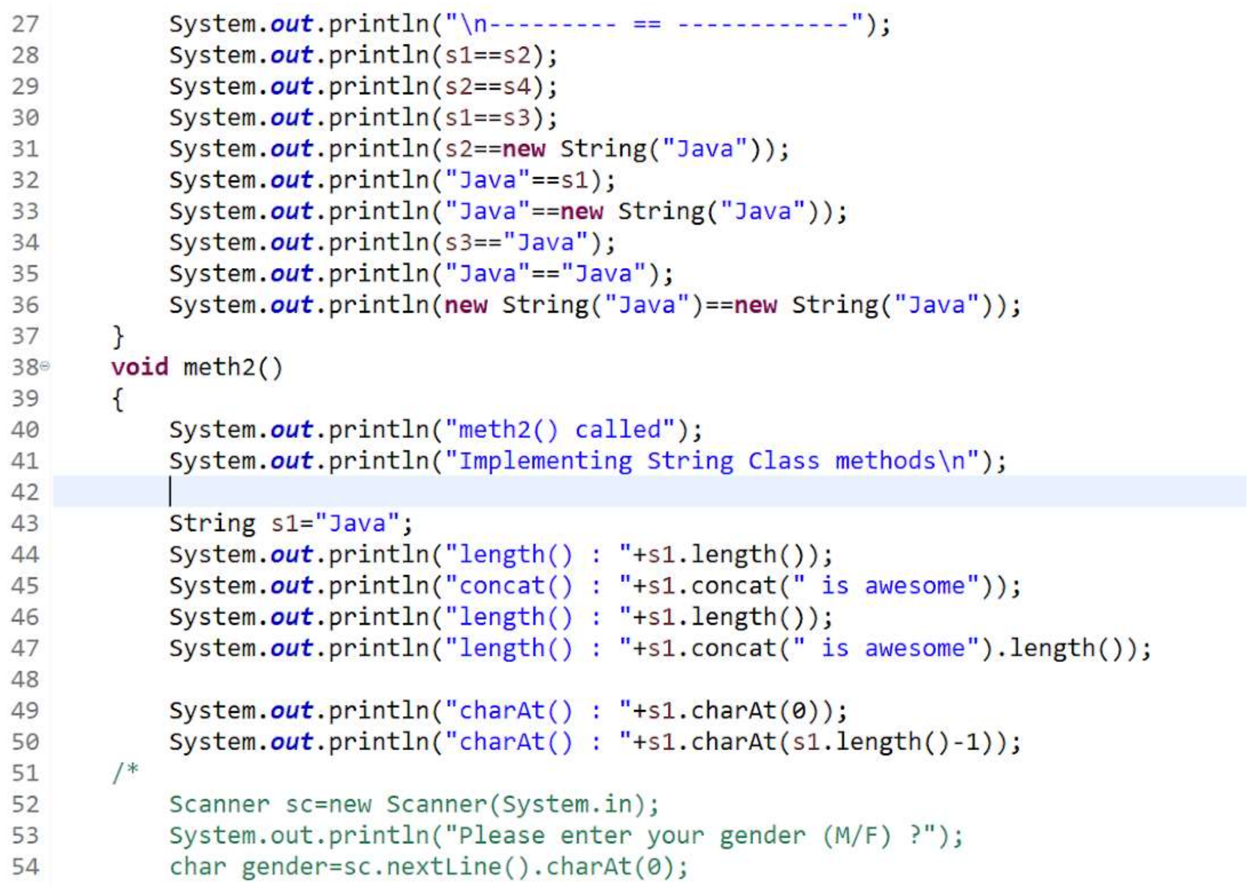
String s=new String("Java"); ⟹ Case 2



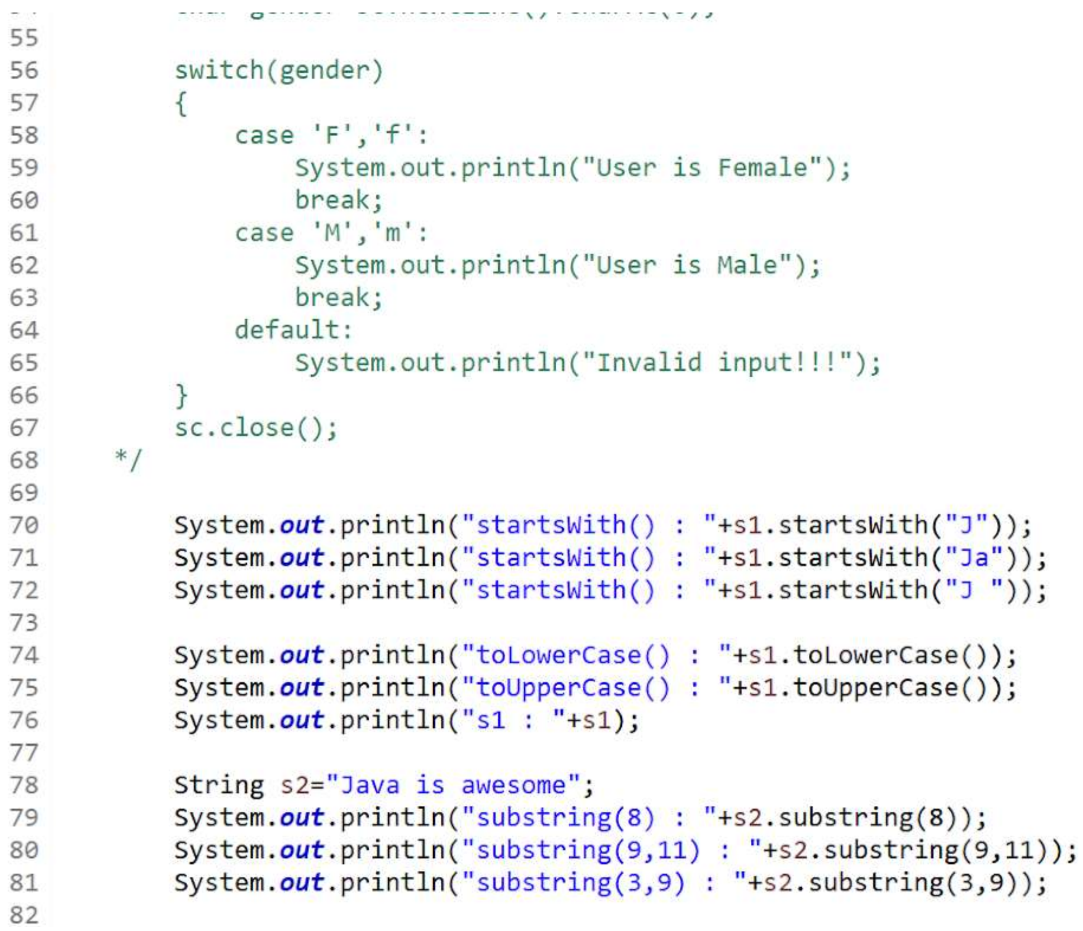
NOTE: In SCP there is NO chance of Duplicate Objects



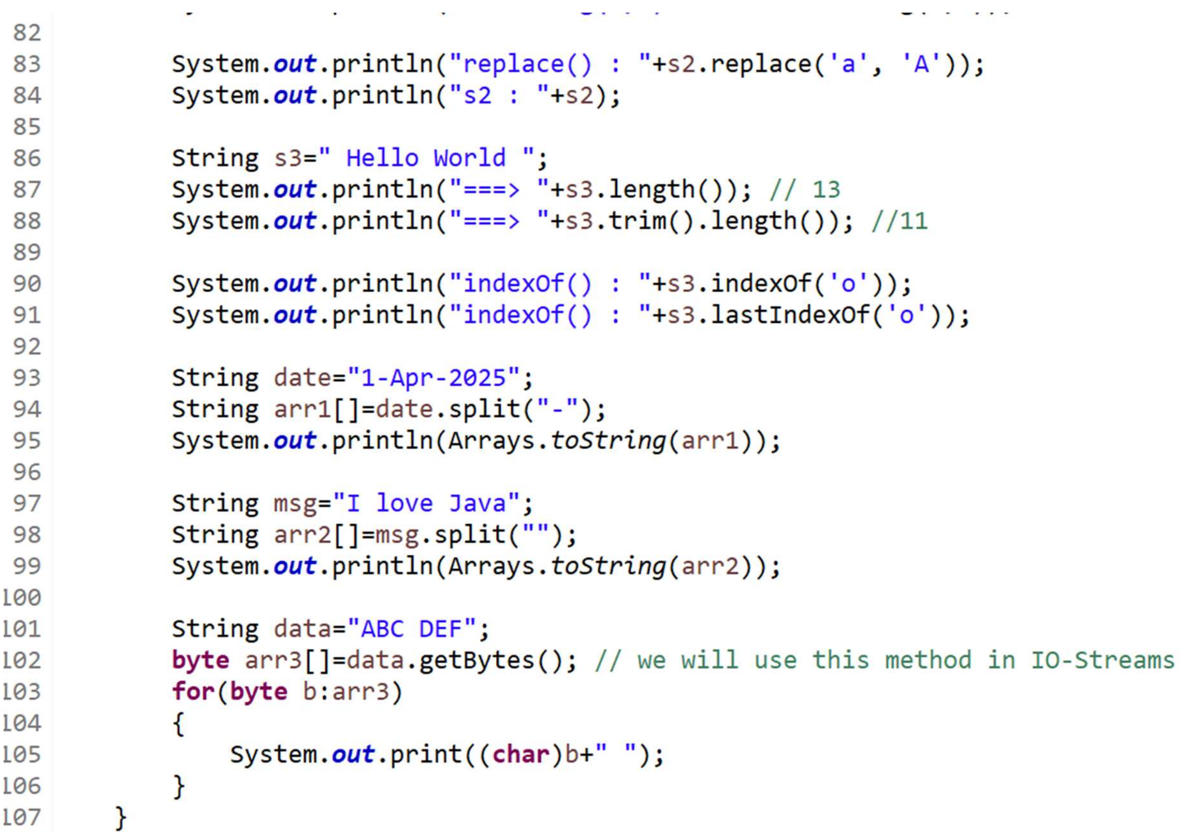
```
import java.util.Arrays;
import java.util.Scanner;

public class ClassA
{

    void meth1()
    {
        System.out.println("meth1() called\n");

        String s1="Java";
        String s2=new String("Java");
        String s3="Java";
        String s4=new String("Java");

        System.out.println("--------equals()---------");
        System.out.println(s1.equals(s2));// t
        System.out.println("Java".equals("Java"));//t
        System.out.println(s2.equals("java"));//f
        System.out.println(s2.equalsIgnoreCase("java"));//t
        System.out.println(s3.equals(new String("java")));//f
        System.out.println(new String("java").equals("java"));//t
        System.out.println(s2.equals(s4));//t

```



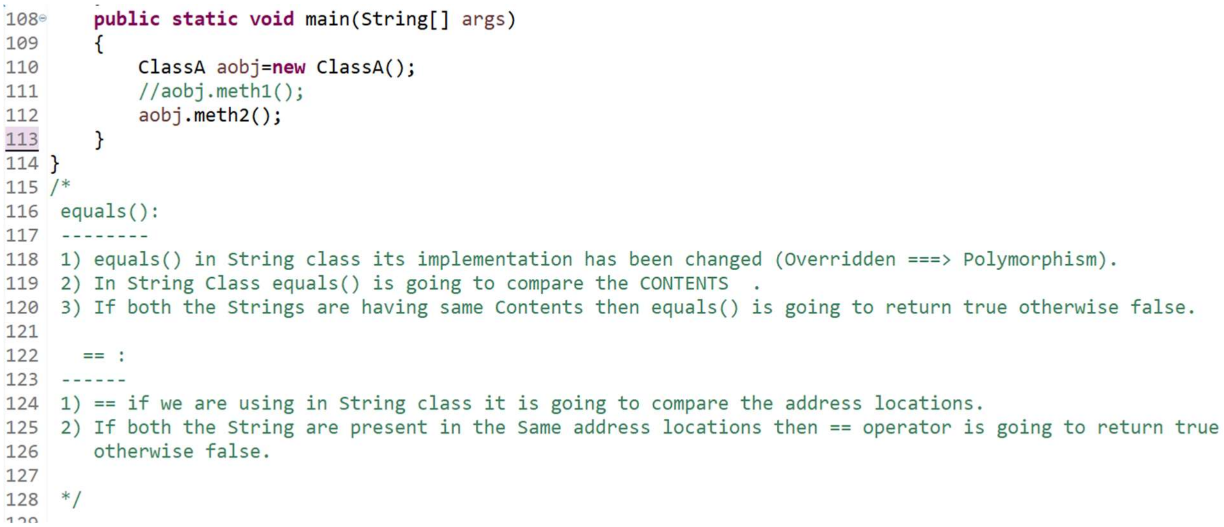
```
        System.out.println("\n--------- == ------------");
        System.out.println(s1==s2);
        System.out.println(s2==s4);
        System.out.println(s1==s3);
        System.out.println(s2==new String("Java"));
        System.out.println("Java"==s1);
        System.out.println("Java"==new String("Java"));
        System.out.println(s3=="Java");
        System.out.println("Java"=="Java");
        System.out.println(new String("Java")==new String("Java"));
    }
    void meth2()
    {
        System.out.println("meth2() called");
        System.out.println("Implementing String Class methods\n");

        String s1="Java";
        System.out.println("length() : "+s1.length());
        System.out.println("concat() : "+s1.concat(" is awesome"));
        System.out.println("length() : "+s1.length());
        System.out.println("length() : "+s1.concat(" is awesome").length());

        System.out.println("charAt() : "+s1.charAt(0));
        System.out.println("charAt() : "+s1.charAt(s1.length()-1));
    /*
        Scanner sc=new Scanner(System.in);
        System.out.println("Please enter your gender (M/F) ?");
        char gender=sc.nextLine().charAt(0);
```



```java

        switch(gender)
        {
            case 'F','f':
                System.out.println("User is Female");
                break;
            case 'M','m':
                System.out.println("User is Male");
                break;
            default:
                System.out.println("Invalid input!!!");
        }
        sc.close();
    */

        System.out.println("startsWith() : "+s1.startsWith("J"));
        System.out.println("startsWith() : "+s1.startsWith("Ja"));
        System.out.println("startsWith() : "+s1.startsWith("J "));

        System.out.println("toLowerCase() : "+s1.toLowerCase());
        System.out.println("toUpperCase() : "+s1.toUpperCase());
        System.out.println("s1 : "+s1);

        String s2="Java is awesome";
        System.out.println("substring(8) : "+s2.substring(8));
        System.out.println("substring(9,11) : "+s2.substring(9,11));
        System.out.println("substring(3,9) : "+s2.substring(3,9));

```



```
        System.out.println("replace() : "+s2.replace('a', 'A'));
        System.out.println("s2 : "+s2);

        String s3=" Hello World ";
        System.out.println("===> "+s3.length()); // 13
        System.out.println("===> "+s3.trim().length()); //11

        System.out.println("indexOf() : "+s3.indexOf('o'));
        System.out.println("indexOf() : "+s3.lastIndexOf('o'));

        String date="1-Apr-2025";
        String arr1[]=date.split("-");
        System.out.println(Arrays.toString(arr1));

        String msg="I love Java";
        String arr2[]=msg.split("");
        System.out.println(Arrays.toString(arr2));

        String data="ABC DEF";
        byte arr3[]=data.getBytes(); // we will use this method in IO-Streams
        for(byte b:arr3)
        {
            System.out.print((char)b+" ");
        }
    }
```



```
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        //aobj.meth1();
        aobj.meth2();
    }
}
/*
 equals():
 --------
 1) equals() in String class its implementation has been changed (Overridden ===> Polymorphism).
 2) In String Class equals() is going to compare the CONTENTS  .
 3) If both the Strings are having same Contents then equals() is going to return true otherwise false.

   == :
 ------
 1) == if we are using in String class it is going to compare the address locations.
 2) If both the String are present in the Same address locations then == operator is going to return true
    otherwise false.

 */
```