Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian

Associate Professor

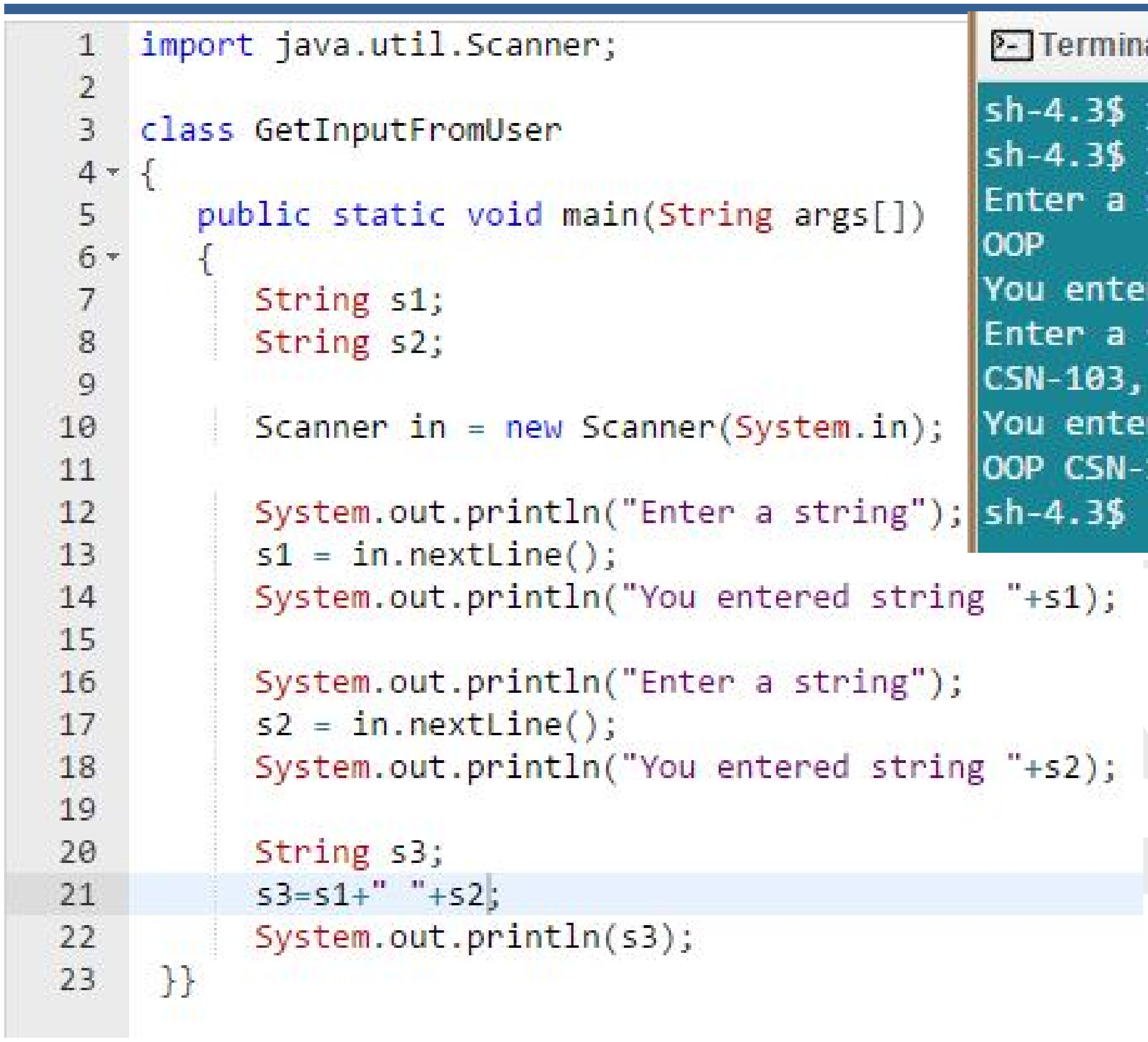
Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

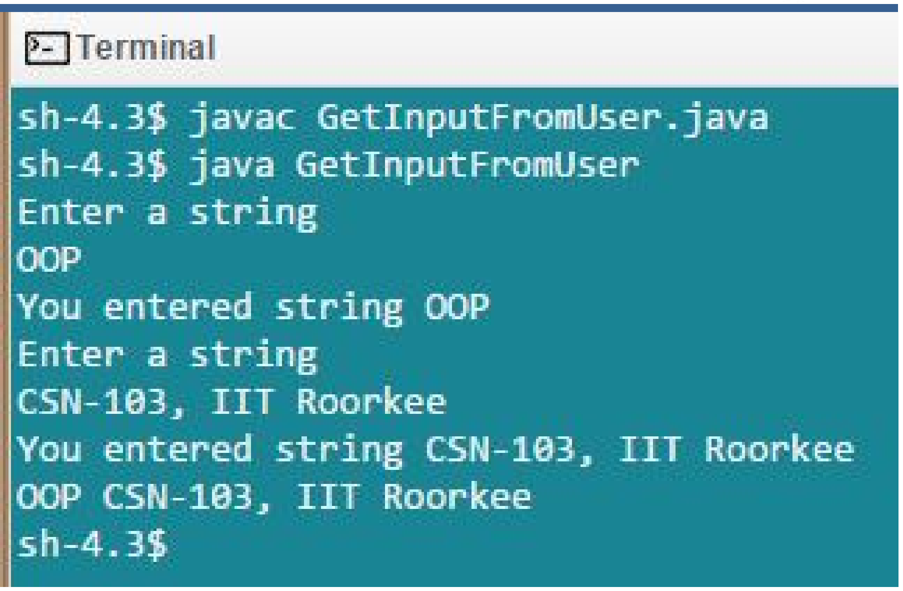
Roorkee 247 667

balarfcs@iitr.ac.in

https://sites.google.com/site/balaiitr/



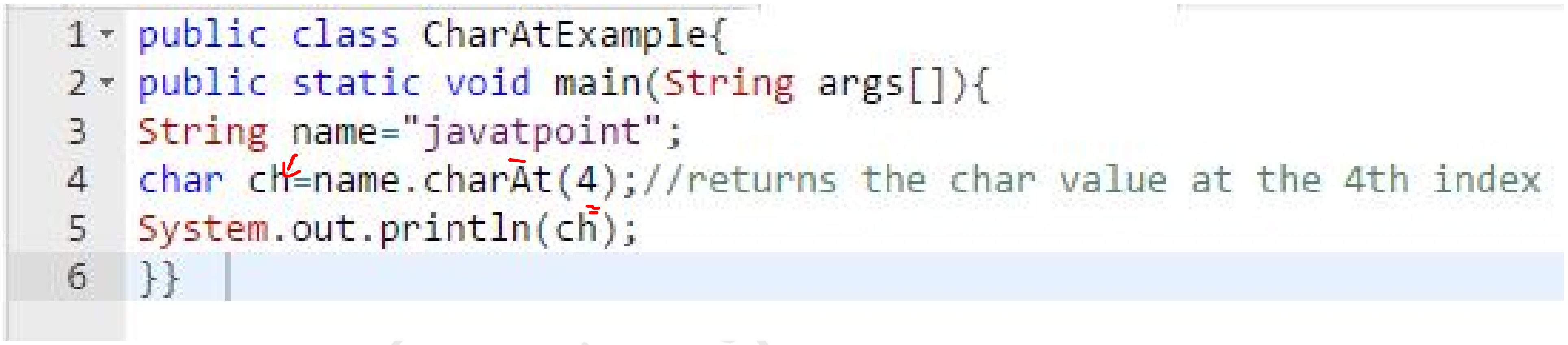
```java
import java.util.Scanner;

class GetInputFromUser
{
    public static void main(String args[])
    {
        String s1;
        String s2;

        Scanner in = new Scanner(System.in);

        System.out.println("Enter a string");
        s1 = in.nextLine();
        System.out.println("You entered string "+s1);

        System.out.println("Enter a string");
        s2 = in.nextLine();
        System.out.println("You entered string "+s2);

        String s3;
        s3=s1+" "+s2;
        System.out.println(s3);
 }}
```



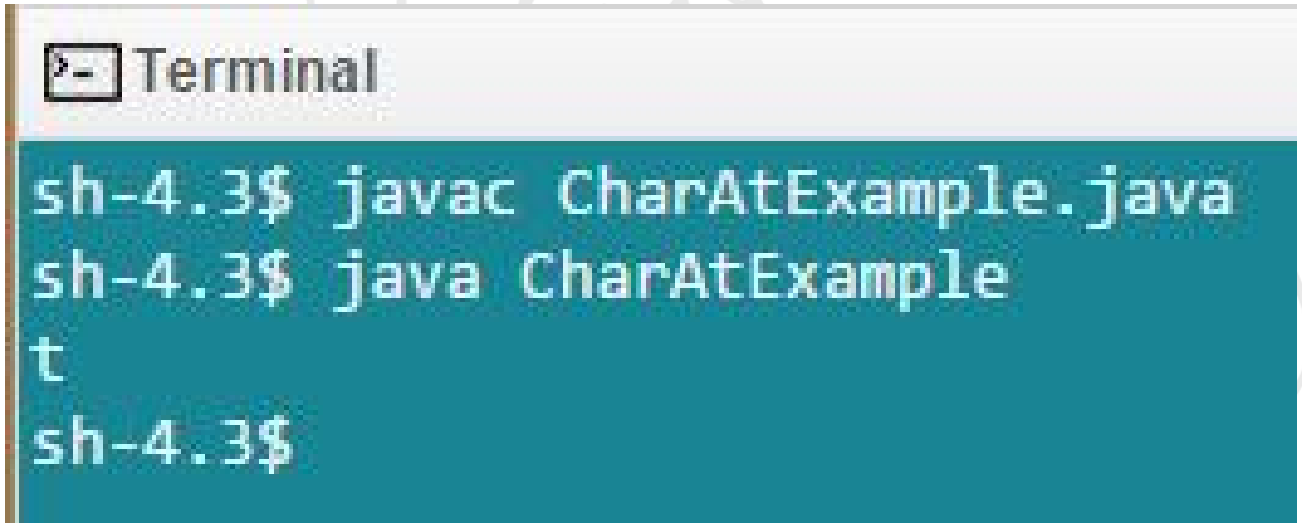
Terminal

```
sh-4.3$ javac GetInputFromUser.java
sh-4.3$ java GetInputFromUser
Enter a string
OOP
You entered string OOP
Enter a string
CSN-103, IIT Roorkee
You entered string CSN-103, IIT Roorkee
OOP CSN-103, IIT Roorkee
sh-4.3$
```

Java String Class methods



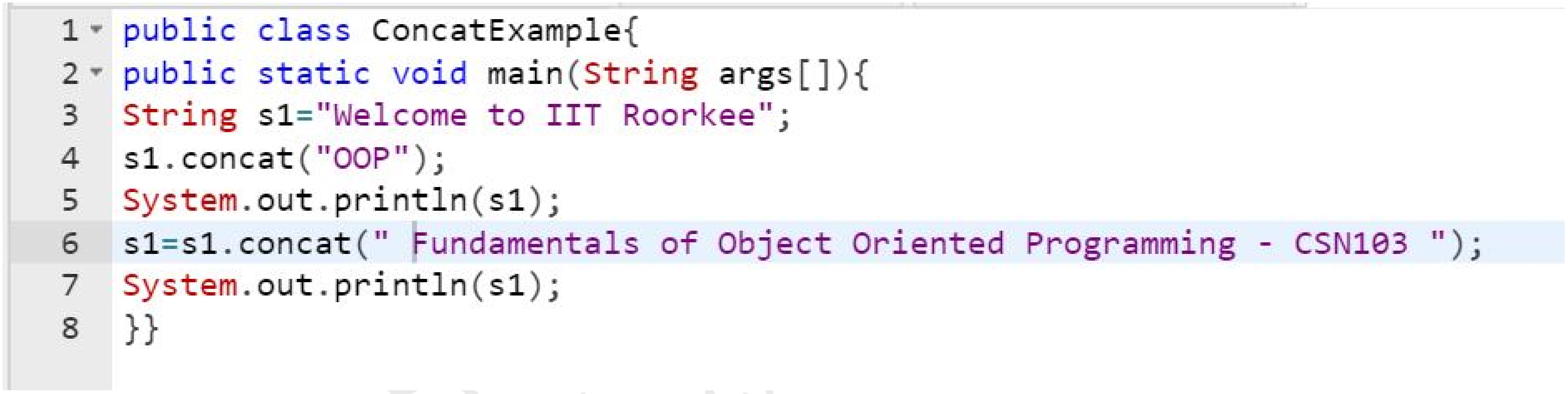
```
public class CharAtExample{
public static void main(String args[]){
String name="javatpoint";
char ch=name.charAt(4);//returns the char value at the 4th index
System.out.println(ch);
}}
```



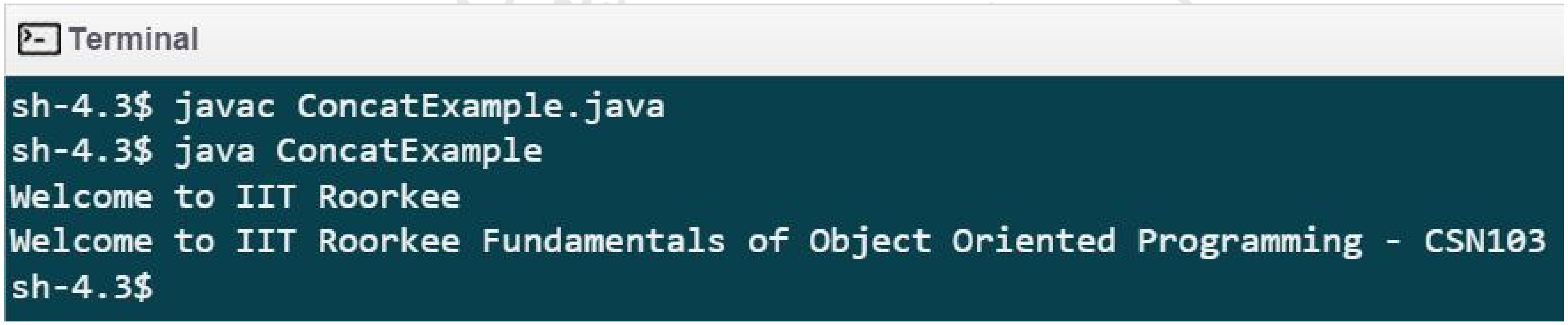
Terminal

```
sh-4.3$ javac CharAtExample.java
sh-4.3$ java CharAtExample
t
sh-4.3$
```

String concat(String str)

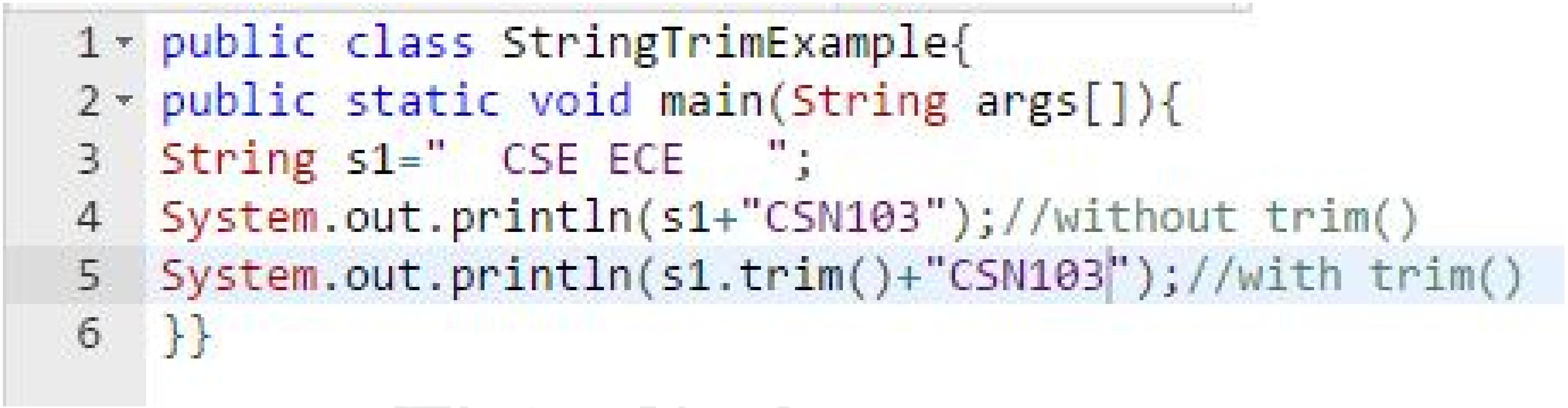


```
public class ConcatExample{
public static void main(String args[]){
String s1="Welcome to IIT Roorkee";
s1.concat("OOP");
System.out.println(s1);
s1=s1.concat(" Fundamentals of Object Oriented Programming - CSN103 ");
System.out.println(s1);
}}
```

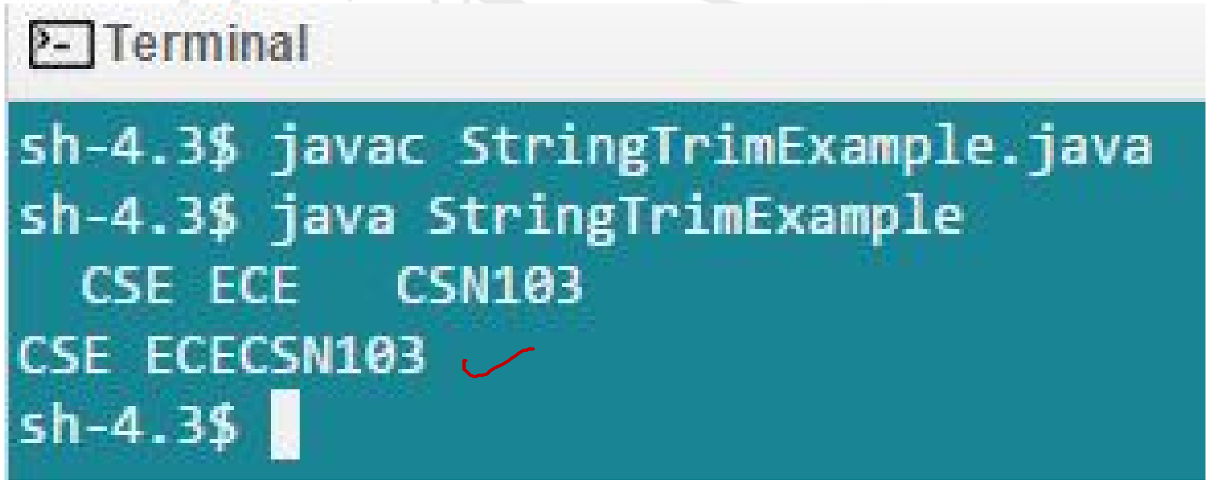


Terminal

```
sh-4.3$ javac ConcatExample.java
sh-4.3$ java ConcatExample
Welcome to IIT Roorkee
Welcome to IIT Roorkee Fundamentals of Object Oriented Programming - CSN103
sh-4.3$
```

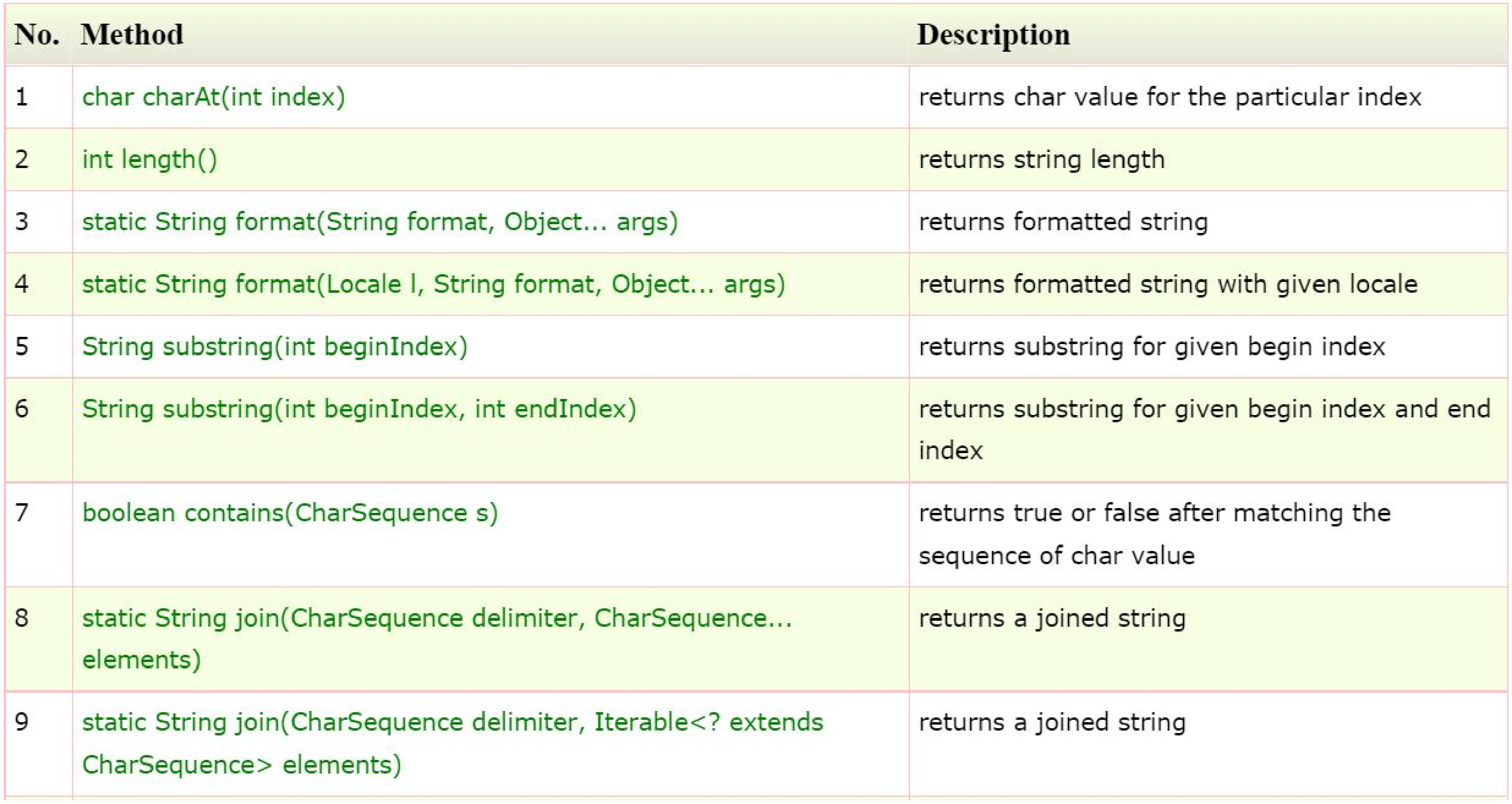


```java
public class StringTrimExample{
public static void main(String args[]){
String s1="  CSE ECE   ";
System.out.println(s1+"CSN103");//without trim()
System.out.println(s1.trim()+"CSN103");//with trim()
}}
```



```
sh-4.3$ javac StringTrimExample.java
sh-4.3$ java StringTrimExample
  CSE ECE   CSN103
CSE ECECSN103 ✓
sh-4.3$
```

http://www.javatpoint.com/java-string



No.	Method	Description
1	char charAt(int index)	returns char value for the particular index
2	int length()	returns string length
3	static String format(String format, Object... args)	returns formatted string
4	static String format(Locale l, String format, Object... args)	returns formatted string with given locale
5	String substring(int beginIndex)	returns substring for given begin index
6	String substring(int beginIndex, int endIndex)	returns substring for given begin index and end index
7	boolean contains(CharSequence s)	returns true or false after matching the sequence of char value
8	static String join(CharSequence delimiter, CharSequence... elements)	returns a joined string
9	static String join(CharSequence delimiter, Iterable<? extends CharSequence> elements)	returns a joined string

10	boolean equals(Object another)	checks the equality of string with object
11	boolean isEmpty()	checks if string is empty
12	String concat(String str) ✓	concatinates specified string
13	String replace(char old, char new)	replaces all occurrences of specified char value
14	String replace(CharSequence old, CharSequence new)	replaces all occurrences of specified CharSequence
15	String trim()	returns trimmed string omitting leading and trailing spaces
16	String split(String regex)	returns splitted string matching regex
17	String split(String regex, int limit)	returns splitted string matching regex and limit
18	String intern()	returns interned string

19	int indexOf(int ch)	returns specified char value index
20	int indexOf(int ch, int fromIndex)	returns specified char value index starting with given index
21	int indexOf(String substring)	returns specified substring index
22	int indexOf(String substring, int fromIndex)	returns specified substring index starting with given index
23	String toLowerCase()	returns string in lowercase.
24	String toLowerCase(Locale l)	returns string in lowercase using specified locale.
25	String toUpperCase()	returns string in uppercase.
26	String toUpperCase(Locale l)	returns string in uppercase using specified locale.

Exercise

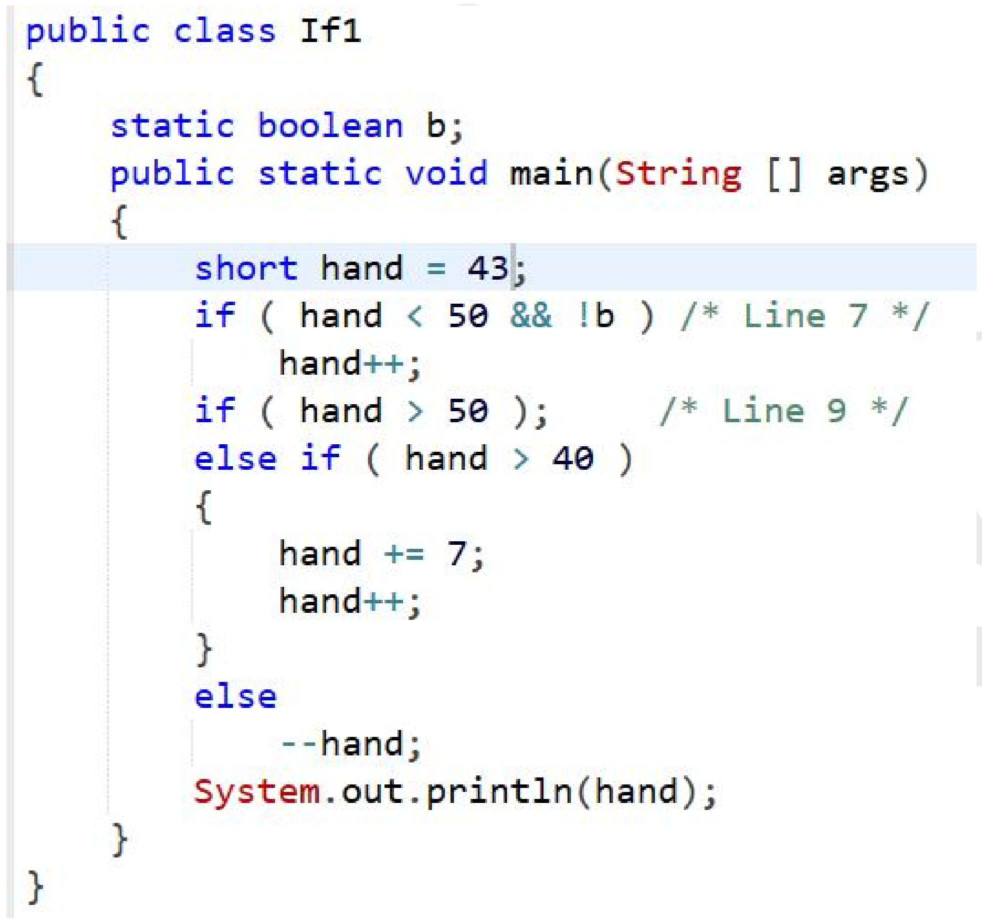
1. Write a JAVA Program to subtract two integers without using – (minus) Operator. **(2 Marks)**

2. Write a JAVA Program using an array to generate first 100 numbers of the following series

   2, 3, 5, 7, 11, 5, 8, 12, 18, 16, 13, 20, 30, 34, ……

   (4 Marks)

3.Write the output of the following JAVA program with proper justification



```java
public class If1
{
    static boolean b;
    public static void main(String [] args)
    {
        short hand = 43;
        if ( hand < 50 && !b ) /* Line 7 */
            hand++;
        if ( hand > 50 );     /* Line 9 */
        else if ( hand > 40 )
        {
            hand += 7;
            hand++;
        }
        else
            --hand;
        System.out.println(hand);
    }
}
```

Classes in JAVA

- In object-oriented programming technique, we design a program using objects and classes.
  - Object is the physical as well as logical entity whereas class is the logical entity only.
- Objects

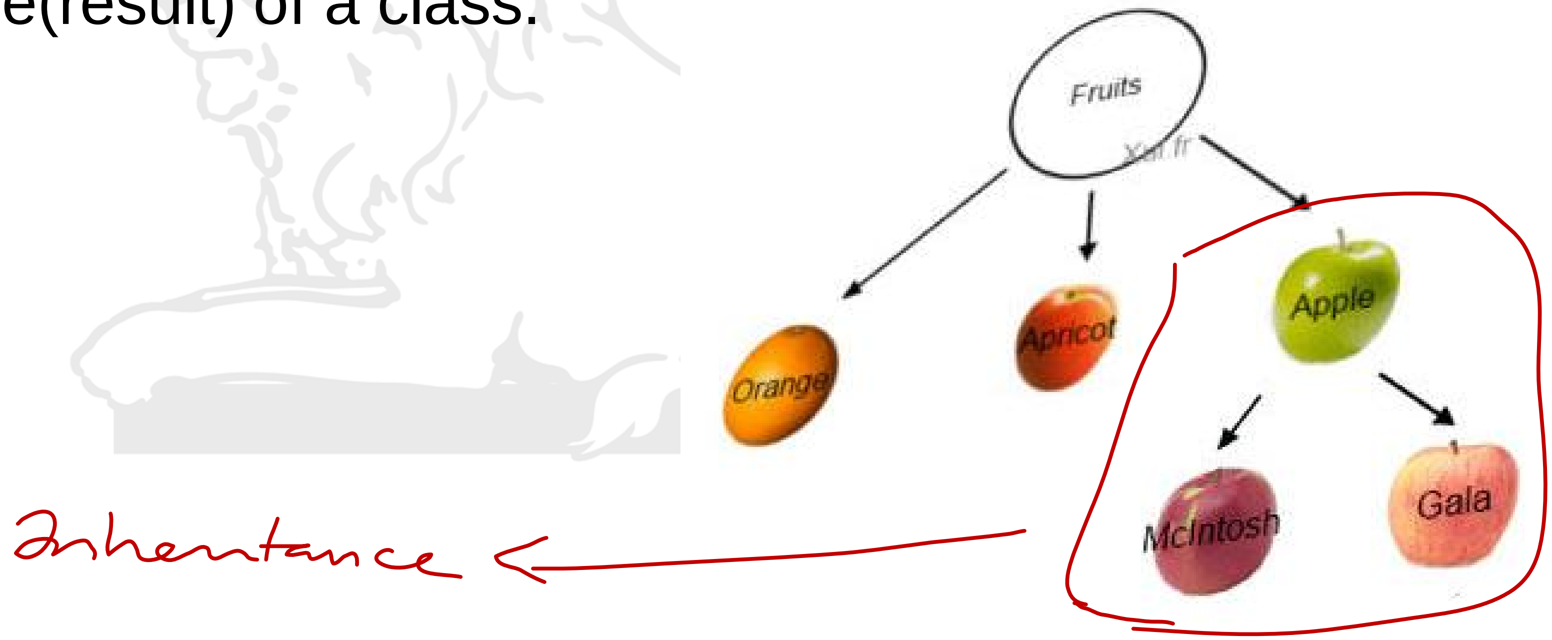
Objects in Java

int a, b;

- An entity that has state and behavior is known as an object e.g. chair, bike, marker, pen, table, car etc. It can be physical or logical (tangible and intangible).
  - The example of intangible object is banking system.
- An object has three characteristics:

- **state:** represents data (value) of an object.
- **behavior:** represents the behavior (functionality) of an object such as deposit, withdraw etc.
- **identity:** Object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. But it is used internally by the JVM to identify each object uniquely.

- For Example: Pen is an object. Its name is Parker, color is Golden etc. known as its state. It is used to write, so writing is its behavior.
- **Object is an instance of a class.** Class is a template or blueprint from which objects are created. So object is the instance(result) of a class.



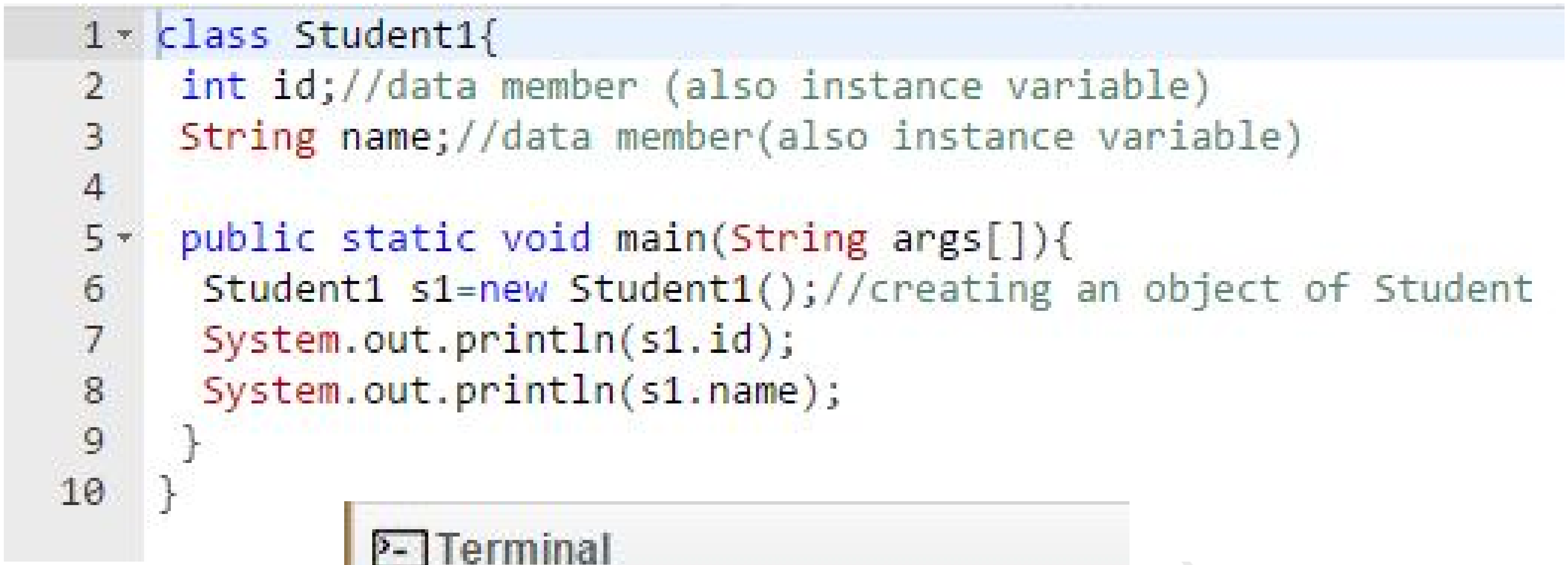
Class in JAVA

- A class is a group of objects that has common properties.
- It is a template or blueprint from which objects are created.

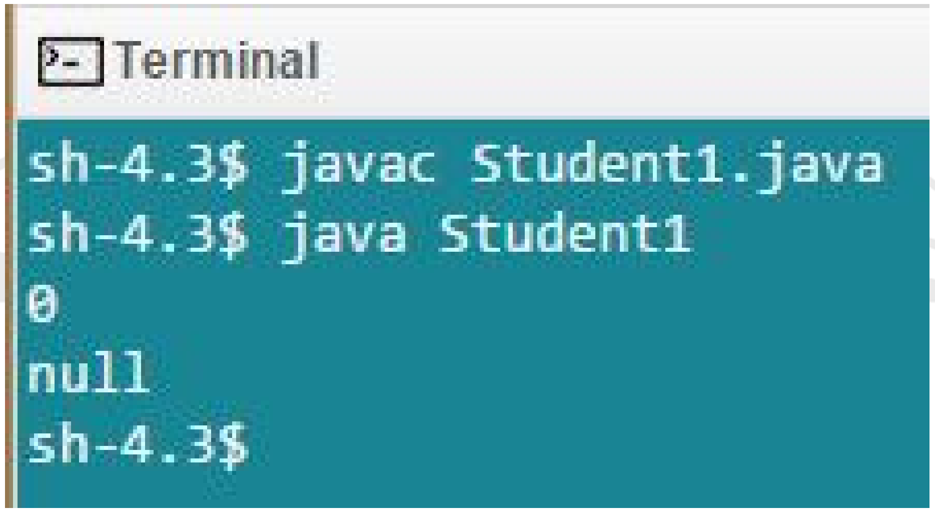
- A class in java can contain:
  - **data member**
  - **method**
  - **constructor**
  - **block**
  - **class and interface**

Syntax to declare a class:

```
class <class_name>{
    data member; //field
    method;
}

// C++ Programmers may note that there is no
//semicolon after the closed braces!
```



```
class Student1{
 int id;//data member (also instance variable)
 String name;//data member(also instance variable)

 public static void main(String args[]){
  Student1 s1=new Student1();//creating an object of Student
  System.out.println(s1.id);
  System.out.println(s1.name);
 }
}
```



```
sh-4.3$ javac Student1.java
sh-4.3$ java Student1
0
null
sh-4.3$
```