Communication Between Two Classes

- In java we can call the member of one class from another class by creating an object.
- It is useful when we need to use common code in every class again and again.
- This is called communication between two classes and can be done in more than one way. You can also say that these object can talk to each other.
- One object sends a message to which the other object replies, or returns a value. They call each other's methods

- **Why do we need communication between classes ?**
  Let's say we have two developers, 'Nilesh' and 'Sachin'. Their manager asked Kiran to write code for a sum of two numbers and sachin to write a code of avg.
  Both of them started to write the code in their own ways. Kiran wrote as below:

```
//@author Nilesh
public class SumLogic {
	int sum(int a, int b) {
		return a+b;
	}
}
```

```
//@author Sachin
public class AvgLogic {
	 int sum(int a, int b) {
		int sum = a+b;
		return sum/2;
	  }
 }
```

- Now, when they submit their code to the manager, he will say that Nilesh wrote the code properly but not Sachin. Why that is so ?
- The reason is that Sachin wrote an additional logic again even through Nilesh had written it earlier.
- Hence it was a waste of time since no reusability feature is used. We can't write logic twice. And since it had already been written by Nilesh , Sachin have used the former's logic instead of repeating it.

- **Now Sachin** rewites his code as shown below :

```
       //@author Sachin
           public class AvgLogic {
                    int avg(int a, int b) {
                    SumLogic sl = new SumLogic ();
                    int sum = sl.sum(a, b);
                    return sum/2;
           }
}
```

- In the industry, a single person cannot write the whole logic.
  Because, it is a mutual effort, we always have to reuse functionalities of each other. I will explain step by step what has happened here.

  ```
        SumLogic sl = new SumLogic ();
  ```

  - Here we have created an object of class SumLogic. It means that we are trying to load SumLogic class by using a keyword called ‘new’.
  - The memory stores all members of class SumLogic [method sum] and s1 knows about address of that location.
  - We can say that sl is eligible to call every thing [members like sum] of SumLogic class. In java, to call members by address we use the dot operator.

- **Consider and study the below examples :-**

Example 1

```
    public class Student_Info {
        int roll_no;
        String name;
        int marks;
        public void getData() throws IOException {
        Scanner sc = new Scanner(System.in);
            BufferedReader br = new BufferedReader( new InputStreamReader(System.in));
              System.out.println("Enter the Roll_No : ");
              roll_no = sc.nextInt();
              System.out.println("Enter the Name : ");
              name = br.readLine();
              System.out.println("Enter the Marks : ");
              marks = sc.nextInt();
        }
        public void showData() {
              System.out.println("Roll No is : "+this.roll_no);
              System.out.println("Name is : "+this.name);
              System.out.println("Marks is : "+this.marks);
        }
  }
```

```
public class Student_Main {
        public static void main(String[] args) throws IOException {
                Student_Info s1 = new Student_Info();
                s1.getData();
                s1.showData();
        }
}
```

Example 2 :-

```
public class A {
        void m6() {
                System.out.println("I am in A-m6");
        }
}
```

```
public class B {
        void m1() {
                System.out.println("I am is B-m1");
        }

        public static void main(String[] args) {
                /*      SCENARIO-1      */
                B bb = new B();
                bb.m1();

                /*      SCENARIO-2      */
                A aa = new A();
                aa.m6();
        }
 }
```

Example 3 :-

```
public class A {
        int a = 10;
        String web = "www.queuecodes.com";
        void show() {
                System.out.println(web);
        }
}
```

```
public class B {
        public static void main(String[] args) {
                A aa = new A();
                System.out.println(aa.a);
                aa.show();
        }
}
```

- **What is System.out.println() ??**
- Here we break down the meaning in three points for you:
    - **System** is class predefined by Sun Microsystem(now Oracle)
    - **out** is the variable declared in System class of type PrintStream which is static
    - **println()** is the method defined in PrintStream class.