```
contact details
============
    haider : syedhyder@ineuron.ai
    nitin     : nitin@ineuron.ai

import java.util.Scanner;
class Two_Sum{
    public  int[] sum(int arr[],int target){
            int[] index;
           looop: for(int i =0 ;i<arr.length;i++){
             for(int j=i+1;j<arr.length-1;j++)
              {
                        int sum=arr[i]+arr[j];
                        if (sum==target){
                              System.out.println("[" + i+ ","+ j+"]");
                          index = new int[2];
                        index[0]=i;
                        index[1] = j;
                              break loop;
                        }
                   }
            return index;
        }
    }
    public static void main(String [] args){
            Two_Sum obj= new Two_Sum();
            int arr[]= new int[10];
            Scanner sc= new Scanner(System.in);
            for(int i=0 ; i<arr.length;i++){
                              arr[i]=sc.nextInt();
            }
            System.out.println("Enter the target element:");
            int target=sc.nextInt();
            obj.sum(arr,target);
    }
}

Why Main method only accept String types of Arguments?

java Test sachin 10 IND
         |
         |
        Test.main(new String[]{"sachin","10","IND"})
If the data is collected in String type, then we can covert the String object to
any type using "Wrapper classes".
class Test{
      public static void main(String[] args){

      }
}


Q>
Example 1:

Input: nums = [2,7,11,15], target = 9
      Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Q> once explain this keyword sir

```
class Student
{
	//instance variables
	String name; int age;

	//local variables are name,age
	Student(String name,int age)
	{
		this.name  =name;
		this.age = age;
	}
}
	Student std = new Student("sachin",10);
```

Note: when name clash occurs between local and instance variable, jvm will give preference only for local variable this concept is called
as Shadowing.
this keyword holds the address of current object, so it resolves the name clash b/w local and instance variables.

From your session on 3rd Nov: Immutability is the main disadvantage of SCP, got it.

But, on 1st Nov, you said creating String literal without newing up is the most efficient way.
Could you please clarify that for me? Thanks in advance!

refer: adhar card application.

Q> What is the difference between static and non-static?

static => common for all the objects of a class.
nonstatic/instance => specific to particular object of a class.

Q>
can you explain about null ...null is belong to which category like object or primitive type or string like that..who is null?
Object variable we call as reference default value is null.
eg: String name = null;
Integer i = null;

Q> Sir Can you discuss about static control flow.

```
class Test{
	static int i;// memory will be located in method area , it will be loaded at the time of loading .class file
	int j;

	static
	{
		System.out.println("Test .class file is loading");
		i =10;
	}
	//instance block will be executed
	{
			j = 20;
```

```
	}
	//constructor will be executed
	public Test()
	{
		j = 100;
	}
}
Test t = new Test();
1 . load the .class file
2. during the loading of .class file memory for static variable will be given with
default value
3. static block will be executed
4. instance block will be executed
5. constructor will be executed.

Q> when to use static method and when to use non static method?
	static method => helper method/ utility methods, if we want to give
facility of the logic to the user
				without creating the objects of its class then make
such methods as "static".
	non-static-method => normally the methods will be non-static as we need the
object to create and then
					get the service of the method.


Q> explain about pass by value and pass by reference

	int a = 10;
	int b = 20;
	add(a,b);//pass by value becoz we are sending primitive type data

			10    20
public void add(int x, int y)
{
	System.out.println("The sum is :: "+(x+y));
}

pass by reference
===============
   Student s1 =new Student();
   Student s2 =new Student();
	validate(s1,s2);//pass by reference as we are sending the object type

public void validate(Student x,Student y)
{

}

Encapsulation
===========
    Process of keeping the data member private and exposing those value to the
outer world through methods is called "encapsulation".

public class Student
{
	private String name;
	public Student(String name){this.name =name;}
	public void setName(String name){this.name =name;}
	public String getName(){return name;}
```

```
}
```