TAKING FAST INPUT IN JAVA

public class Demo{

public static void main( String[] args){

System.out.println( “”) ;

}

}

println is amethod of which class ?

It belong to PrintStream Class

So I need to create object of print stream

The pbject is already created

Out object,

It is created as static variable inside the System class

It out is static object inside System class , so I can easily use it

There is also a varible in like out of PrintStream class

int num = System.in.read()

println( num )

The point here is, it reads only 1 character even if we give input anything + it reads the ASCII value

To read multiple chacetr use loop

So java give special class

BufferedReader

InputStreamreader in = new InputStreamReader(System.in) ;

BufferedReadre bf = new BufferedReader(in) ;

int num = Integer.parseInt(bf.readLine() ) ;

println( num ) ;

bf.close() ;

// close the resources after opening them

BufferedReader bf = new BufferedReader( new InputStreamReader( System.in)) ;

After reading by bf.readLine you have to parse the content to Integr.parseInt() , Long.parseLong() , Double.parseDouble() ;

BUFFERED READER TAKES AN RESOURCE , LIKE FILE OR SOMTHIGN SAME FR BUFFEREDWRITER

// Creates a FileWriter

FileWriter file = new FileWriter(String name);

// Creates a BufferedWriter

BufferedWriter buffer = new BufferedWriter(file);

BufferedWriter bw = new BufferedWriter( new OutputStreamWriter( System.out)) ;

PRINTING BY BUFFEREDWRITER

import java.io.BufferedWriter;

import java.io.FileWriter;

import java.io.IOException;

***public class BufferedWriterExample {***

***public static void main(String[] args) {***

***String filePath = "example.txt"; // The path to the file where data will be written.***

***String data = "This is an example of using BufferedWriter.\nBufferedWriter can write text efficiently.";***

***try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath))) {***

***writer.write(data);***

***writer.newLine(); // Adds a new line to the file.***

***writer.write("This is another line of text.");***

***} catch (IOException e) {***

***e.printStackTrace();***

***}***

***}***

***}***

In scanner ,using sc.nextLine() after sc.nextInt() is an error

Sc.nextInt()

Sc.nextLine() ;

Sc.nextLine() ;