**Day11 – 21st June 2025**

**TASK1 - Run the below code and see the file with the given name created or not..**

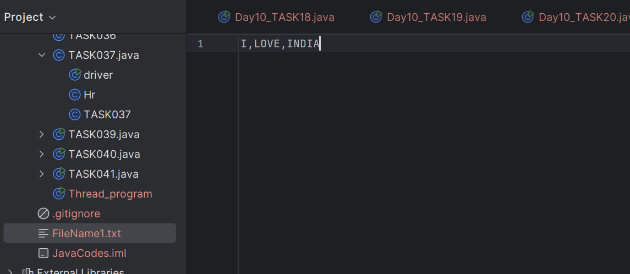
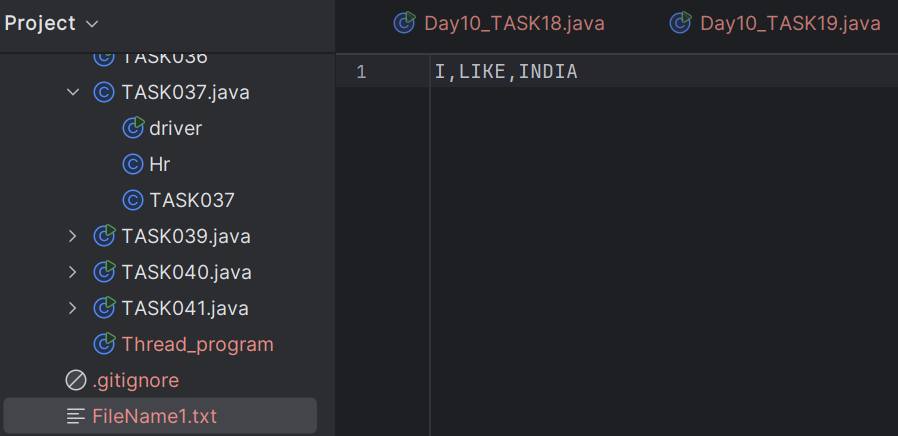
**Run it again with “I like India” instead of “I love India”.. And see the file …**

import java.io.\*;  
import java.io.IOException;  
public class Day11\_TASK1  
{  
 public static void main(String args[])  
 {  
 File f1=new File("FileName1.txt");  
 FileOutputStream outfile = null;  
 byte Text[] = {'I', ',','L','O','V','E',',','I','N','D','I','A'};  
 try  
 {  
 outfile = new FileOutputStream(f1);  
 outfile.write(Text);  
 }  
 catch(IOException e)  
 {  
 System.*out*.println(e);  
 System.*exit*(-1);  
 }  
 System.*out*.println("Write Byte");  
 System.*out*.println("Thank You...!!!");  
 }  
}

**Output**

**Write Byte**

**Thank You...!!!**



**TASK2 - Write a program which reads byte from file.**

import java.io.\*;  
public class Day11\_TASK2  
{  
 public static void main(String args[])  
 {  
 FileInputStream infile = null;  
 int b;  
 try  
 {  
 infile = new FileInputStream("FileName01.txt");  
 while((b = infile.read()) != -1)  
 {  
 System.*out*.println((char)b);  
 }  
 infile.close();  
 }  
 catch(IOException e)  
 {  
 System.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
 }  
}

**Output**

**Sorry..!! File Not Found...!!!**

import java.io.\*;  
public class Day11\_TASK2  
{  
 public static void main(String args[])  
 {  
 FileInputStream infile = null;  
 int b;  
 try  
 {  
 infile = new FileInputStream("FileName1.txt");  
 while((b = infile.read()) != -1)  
 {  
 System.*out*.println((char)b);  
 }  
 infile.close();  
 }  
 catch(IOException e)  
 {  
 System.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
 }  
}

**Output**

**I , L I K E , I N D I A**

**TASK3 - Taking input from the user and writing on the file… Create a file and see the output…**

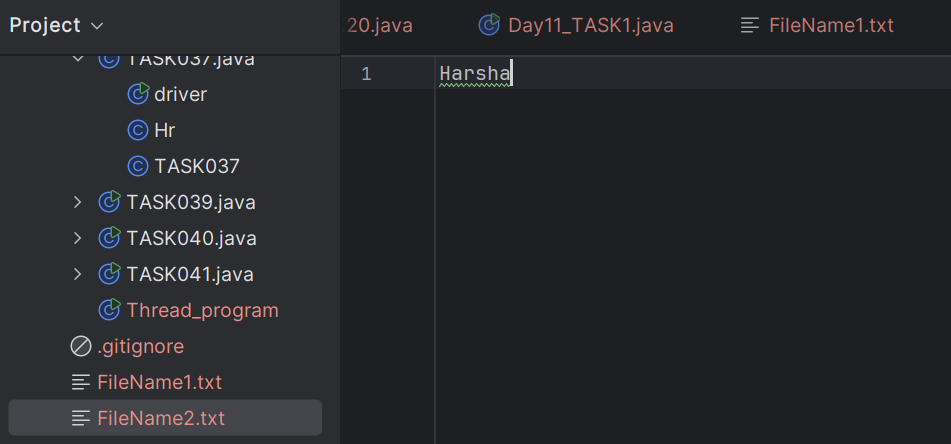
import java.io.\*;  
import java.util.\*;  
public class Day11\_TASK3  
{  
 public static void main(String args[]) {  
 FileOutputStream outfile = null;  
//String s=args[0]; // to input string from command line  
 Scanner sc=new Scanner(System.*in*);  
 String s=sc.nextLine();  
 byte b1[] = s.getBytes();  
 try  
 {  
 outfile = new FileOutputStream("FileName2.txt");  
 outfile.write(b1);  
 }  
 catch(IOException e)  
 {  
 System.*out*.println(e);  
 System.*exit*(-1);  
 }  
 System.*out*.println("Write Byte");  
 System.*out*.println("Thank You...!!!");  
 }  
}

**Output**

**Harsha**

**Write Byte**

**Thank You...!!!**



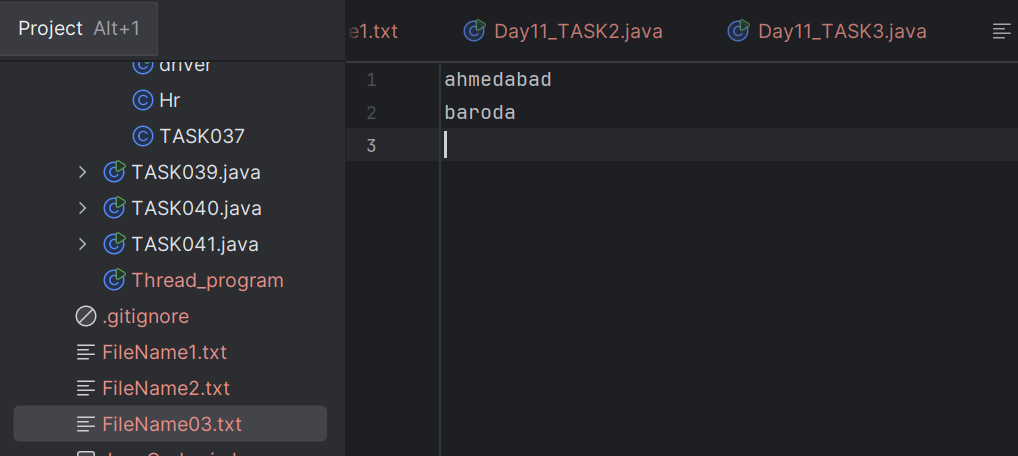
**Reading/writing characters**

**TASK4 - Write a program which creates file and writes character into that file.**

import java.io.\*;  
class Day11\_TASK4  
{  
 public static void main(String args[])  
 {  
 File f1 = new File("FileName03.txt");  
 FileWriter fw = null;  
 try {  
 fw = new FileWriter(f1);  
 fw.write("ahmedabad \n");  
 fw.write(" baroda \n");  
 fw.close();  
 } catch (FileNotFoundException e) {  
 System.*out*.println("Sorry..!! File Not Found...!!!");  
 } catch (IOException e) {  
 System.*out*.println(e.getMessage());  
 }  
 System.*out*.println("Write Operation done");  
 }  
  
}

**Output**

**Write Operation done**



**TASK5 - Write a program which reads character from file.**

import java.io.\*;  
class Day11\_TASK5  
{  
public static void main(String args[])  
{  
 FileReader fr =null;  
 try  
 {  
 fr = new FileReader("FileName03.txt");  
 int ch;  
 while((ch = fr.read()) != -1)  
 {  
 System.*out*.print((char)ch);  
 }  
 System.*out*.println("Reading complete");  
 fr.close();  
 }  
 catch(FileNotFoundException e)  
 {  
 System.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
 catch(IOException e)  
 {  
 System.*out*.println(e.getMessage());  
 }  
}  
}

**Output**

**ahmedabad**

**baroda**

**Reading complete**

**TASK6 - Write a program to read one byte at a time from a file and copy it into another file immediately**.

import java.io.\*;  
class Day11\_TASK6  
{  
 public static void main(String args[])  
 {  
 try  
 {  
 byte b=0;  
 FileInputStream infile = new FileInputStream("FileName1.txt");  
 FileOutputStream outfile = new FileOutputStream("FileName03.txt");  
 int byteread;  
 while ((byteread = infile.read()) != -1) {  
 outfile.write(byteread);  
 }  
 System.*out*.println("Byte Copied From in.txt to out.txt FIle ");  
 }  
 catch(FileNotFoundException e)  
 {  
 System.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
 catch(IOException e)  
 {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

**Output**

**Byte Copied From in.txt to out.txt File**

**Same text that is in FileName1.txt**

**TASK7 - Write a program to merge two files in third file.**

import java.io.\*;  
class Day11\_TASK7  
{  
 public static void main(String args[])  
 {  
 try  
 {  
 FileInputStream file1 = new FileInputStream("FileName1.txt");  
 FileInputStream file2 = new FileInputStream("FileName03.txt");  
 SequenceInputStream file3 = new SequenceInputStream(file1, file2);  
  
 BufferedInputStream br1 = new BufferedInputStream(file3);  
 BufferedOutputStream br2 = new BufferedOutputStream(new FileOutputStream("FileName4.txt"));  
 int ch;  
 while((ch = br1.read())!=-1)  
 {  
 br2.write((char)ch);  
 }  
 br1.close();  
 br2.close();  
 file1.close();  
 file2.close();  
 System.*out*.println("Merge Two File Sucessfully ");  
 }  
 catch(IOException e)  
 {  
 System.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
 }  
}

**Output**

**Merge Two File Sucessfully**

I LOVE INDIA

ahmedabad

baroda

#### Notes:

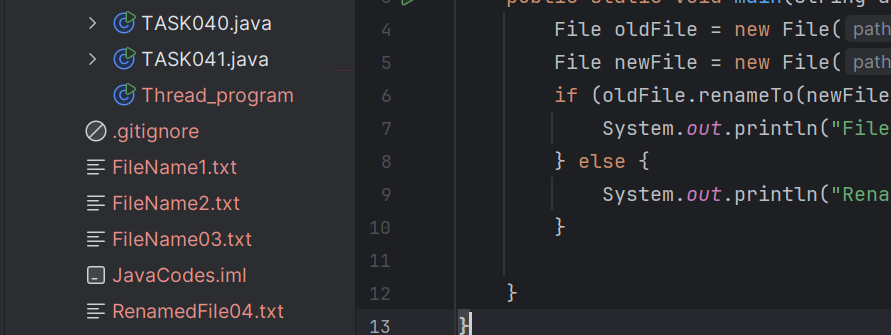
* Merges contents of two files using SequenceInputStream.
* Buffered streams make reading/writing efficient.
* Appends content in sequence (not line-by-line logic).

**TASK8 - Write an application to rename a file. Use the renameTo() method of File to  accomplish**

import java.io.\*;  
class Day11\_TASK8 {  
 public static void main(String args[]) {  
 File oldFile = new File("FileName4.txt");  
 File newFile = new File("RenamedFile04.txt");  
 if (oldFile.renameTo(newFile)) {  
 System.*out*.println("File renamed successfully.");  
 } else {  
 System.*out*.println("Rename failed.");  
 }  
  
 }  
}

**Output**

**File renamed successfully.**



**TASK9 - Lambda Expressions to Check String Length**

interface Checker {  
 boolean check(String str);  
}  
  
public class Day11\_TASK9 {  
 public static void main(String[] args) {  
 Checker checker = (str) -> str.length() > 5;  
 System.*out*.println(checker.check("Hello")); // false  
 System.*out*.println(checker.check("Greetings")); // true  
 }  
}

**Output**

**False**

**True**

**TASK10 - Write a code to reverse a string.**

import java.util.function.Function;  
  
public class Day11\_TASK10 {  
 public static void main(String[] args) {  
 String input = "Lambda";  
  
 // Lambda expression to reverse a string using StringBuilder  
 Function<String, String> reverse = str -> new StringBuilder(str).reverse().toString();  
  
 String result = reverse.apply(input);  
  
 System.*out*.println("Original: " + input);  
 System.*out*.println("Reversed: " + result);  
 }  
}

**Output**

**Original: Lambda**

**Reversed: adbmaL**

**TASK11 – Another method of reversing a string**

interface MyInterface {  
  
 // abstract method  
 String reverse(String n);  
}  
  
public class Day11\_TASK11 {  
  
 public static void main( String[] args ) {  
  
 // declare a reference to MyInterface  
 // assign a lambda expression to the reference  
 MyInterface ref = (str) -> { String result = "";  
 for (int i = str.length()-1; i >= 0 ; i--)  
 result += str.charAt(i);  
 return result;  
 };  
  
 // call the method of the interface  
 System.*out*.println("Lambda reversed = " + ref.reverse("Lambda"));  
 }  
  
}

**Output**

**Lambda reversed = adbmaL**

**TASK12 – Write a program to create an array list with 5 friends names.**

import java.util.ArrayList;  
public class Day11\_TASK12  
{  
 public static void main(String[] args)  
 {  
 ArrayList<String> names = new ArrayList<>();  
 names.add("Harsha");  
 names.add("Arun");  
 names.add("Kumar");  
 names.add("Teddy");  
 names.forEach(name-> System.*out*.println(name));  
 }  
}

**Output**

**Harsha**

**Arun**

**Kumar**

**Teddy**

**TASK13 – Write a program to create a List of 5 friends names (first name and last name)**

import java.util.ArrayList;  
public class Day11\_TASK13 {  
 public static void main(String[] args) {  
 ArrayList<String> fullnames = new ArrayList<>();  
 fullnames.add("Harsha Bheemisetty");  
 fullnames.add("Arun Ramaji");  
 fullnames.add("Kumar Arun");  
 fullnames.add("Teddy Ramaji");  
 fullnames.forEach(System.*out*::println);  
 }  
}

**Output**

**Harsha Bheemisetty**

**Arun Ramaji**

**Kumar Arun**

**Teddy Ramaji**

**TASK14 – FILTER, MAPS, SORT with Streams**

import java.util.\*;  
public class Day11\_TASK14 {  
 public static void main(String[] args) {  
 List<String> fullNames = Arrays.*asList*(  
 "Arun Ramaji",  
 "Harsha Bheemisetty",  
 "Anusha Bheemisetty",  
 "Neelima Ramaji",  
 "Lavany Ramaji");  
  
 fullNames.stream()  
 .filter(name -> name.startsWith("H"))  
 .map(String::toUpperCase)  
 .sorted()  
 .forEach(System.*out*::println);  
 }  
}

**Output**

**HARSHA BHEEMISETTY**

import java.util.ArrayList;  
import java.util.List;  
  
public class Day11\_TASK14\_1 {  
  
 // create an object of list using ArrayList  
 static List<String> *places* = new ArrayList<>();  
  
 // preparing our data  
 public static List getPlaces(){  
  
 // add places and country to the list  
 *places*.add("Nepal, Kathmandu");  
 *places*.add("Nepal, Pokhara");  
 *places*.add("India, Delhi");  
 *places*.add("USA, New York");  
 *places*.add("Africa, Nigeria");  
  
 return *places*;  
 }  
  
 public static void main( String[] args ) {  
  
 List<String> myPlaces = *getPlaces*();  
 System.*out*.println("Places from Nepal:");  
  
 // Filter places from Nepal  
 myPlaces.stream()  
 .filter((p) -> p.startsWith("Nepal"))  
 .map((p) -> p.toUpperCase())  
 .sorted()  
 .forEach((p) -> System.*out*.println(p));  
 }  
  
}

**Output**

**Places from Nepal:**

**NEPAL, KATHMANDU**

**NEPAL, POKHARA**

**TASK15 - Wap to accept or create a list of 5 integers and display the squares of each ..**

#### Square Numbers using map() and collect()

import java.util.\*;  
import java.util.stream.Collectors;  
public class Day11\_TASK15 {  
 public static void main(String[] args) {  
 List<Integer> numbers = Arrays.*asList*(2, 4, 6, 8, 10);  
  
 List<Integer> squares = numbers.stream()  
 .map(num -> num \* num)  
 .collect(Collectors.*toList*());  
  
 System.*out*.println("Squares: " + squares);  
 }  
}

**Output**

**Squares: [4, 16, 36, 64, 100]**

**TASK16 - Write a code to create an array list and filter the values which are odd numbers and display them..**

import java.util.\*;  
import java.util.stream.Collectors;  
public class Day11\_TASK16 {  
 public static void main(String[] args) {  
 List<Integer> numbers = Arrays.*asList*(11, 22, 33, 44, 55, 77, 12, 13);  
  
 List<Integer> oddNumbers = numbers.stream()  
 .filter(num -> num % 2 != 0)  
 .collect(Collectors.*toList*());  
  
 System.*out*.println("Odd Numbers: " + oddNumbers);  
 }  
}

**Output**

**Odd Numbers: [11, 33, 55, 77, 13]**

**TASK17 – Write a program to create an array list to remove duplicate values from the List.**

import java.util.\*;  
import java.util.stream.Collectors;  
public class Day11\_TASK17 {  
 public static void main(String[] args) {  
 List<Integer> numbers = Arrays.*asList*(10, 20, 10, 30, 20, 40, 50, 40);  
  
 List<Integer> uniqueNumbers = numbers.stream()  
 .distinct()  
 .collect(Collectors.*toList*());  
  
 System.*out*.println("Unique Numbers: " + uniqueNumbers);  
 }  
}

**Output**

**Unique Numbers: [10, 20, 30, 40, 50]**

**TASK18 – Write a program to run a loop / iterate()  and limit it to 20 values (1 to 20)**

**While displaying use for each to limit till 10 numbers.**

import java.util.stream.Stream;  
public class Day11\_TASK18 {  
 public static void main(String[] args) {  
 Stream<Integer> numbers = Stream.*iterate*(1, n -> n + 1).limit(20);  
  
 numbers.limit(10).forEach(System.*out*::println);  
 }  
}

**Output**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**10**

**TASK19 – Write a program to create an array List skip 15 numbers and print the output using foreach loop.**

import java.util.stream.Stream;  
public class Day11\_TASK19 {  
 public static void main(String[] args) {  
 Stream<Integer> numbers = Stream.*iterate*(1, n -> n + 1).limit(20);  
  
 numbers.skip(15).forEach(System.*out*::println);  
 }  
}

**Output**

**16**

**17**

**18**

**19**

**20**

#### **TASK20 –** Reduce (sum, max, concatenation)

import java.util.Arrays;  
import java.util.List;  
import java.util.Optional;  
public class Day11\_TASK20 {  
 public static void main(String[] args) {  
  
 List<Integer> numbers = Arrays.*asList*(1, 2, 3, 4, 5);  
  
 Optional<Integer> sum = numbers.stream().reduce((x, y) -> x + y);  
 System.*out*.println("Sum of all elements: " + sum.orElse(0));  
  
  
 Optional<Integer> max = numbers.stream().reduce(Integer::*max*);  
 System.*out*.println("Maximum element: " + max.orElse(0));  
  
 List<String> strings = Arrays.*asList*("Hello", " ", "world", "!");  
 Optional<String> concatenatedString = strings.stream().reduce((x, y) -> x + y);  
 System.*out*.println("Concatenated string: " + concatenatedString.orElse(""));  
 }  
}

**Output**

**Sum of all elements: 15**

**Maximum element: 5**

**Concatenated string: Hello world!**