Task 1:

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 …

Code:

package June21;  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
public class Task1  
{  
 public static void main(String[] args)  
 {  
 File f1=new File("FileName01.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...!!!

Process finished with exit code 0

Task 2:

Try this code to see the output …

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

import java.io.\*;

public class ReadingByte

{

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...!!!");

}

}

}

Code:

package June21;  
  
import java.io.\*;  
public class ReadingByte  
{  
 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:

I

L

O

V

E

I

N

D

I

A

Process finished with exit code 0

**Task 4:**

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

import java.io.\*;

Class CharacterWrite {

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!!”);

}

}

Code:

package June21;  
  
import java.io.\*;  
public class CharacterWrite {  
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!!

Process finished with exit code 0

Task 5:

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

import java.io.\*;

Class Readchar

{

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());

}

}

}

Code:

package June21;  
  
import java.io.\*;  
public class Readchar  
{  
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

Process finished with exit code 0

Task 6:

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

import java.io.\*;

Class CopyByte

{

public static void main(String args[])

{

try

{

byte b=0;

FileInputStream infile = new FileInputStream("NewFile01.txt");

FileOutputStream outfile = new FileOutputStream("NewFile05.txt");

Initialize byteread here….

while(byteread != -1)

{

b = (byte)infile.read();

outfile.write(b);

}

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());

}

}

}

Code:

package June21;  
  
import java.io.\*;  
public class CopyByte  
{  
public static void main(String[] args)  
{  
 try  
 {  
 byte b=0;  
 FileInputStream infile = new FileInputStream("FileName01.txt");  
 FileOutputStream outfile = new FileOutputStream("FileName04.txt");  
  
 FileReader byteread = new FileReader("FileName01.txt");  
 while( byteread.read() != -1 )  
 {  
 b = (byte)infile.read();  
 outfile.write(b);  
 }  
 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

Process finished with exit code 0

Task 7:

Merging two files to 3rd file..

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

import java.io.\*;

classFileMergeDemo

{

public static void main(String args[])

{

try

{

FileInputStream file1 = new FileInputStream("NewFile01.txt");

FileInputStream file2 = new FileInputStream("NewFile02.txt");

SequenceInputStream file3 = new SequenceInputStream(file1, file2);

BufferedInputStream br1 = new BufferedInputStream(file3);

BufferedOutputStream br2 = new BufferedOutputStream(System.out);

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...!!!");

}

}

}

Code:

package June21;  
  
import java.io.\*;  
import java.nio.file.Files;  
import java.nio.file.Paths;  
  
public class FileMergeDemo  
{  
 public static void main(String[] args)  
 {  
 try  
 {  
 FileInputStream file1 = new FileInputStream("FileName01.txt");  
 FileInputStream file2 = new FileInputStream("FileName04.txt");  
  
 SequenceInputStream file3 = new SequenceInputStream(file1, file2);  
  
 BufferedInputStream br1 = new BufferedInputStream ( file3 );  
 BufferedOutputStream br2 = new BufferedOutputStream(Files.*newOutputStream*(Paths.*get*("FileName05.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

Process finished with exit code 0

Task 8:

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

/\*this task. The first command line argument is the old filename and the second is  the newfilename.

\*/

import java.io.\*;

classFileRenameDemo

{

public static void main(String args[])

{

File f1 = new File(args[0]);

File f2 = new File(args[1]);

f1.renameTo(f2);

System.out.println("Rename File " +f1+" To "+f2+" Sucessfully "); }

}

Output :

Javac FileRenameDemo.java

Java FileRenameDemo input1.txt abc.txt

Code:

package June21;  
import java.io.\*;  
public class FileRenameDemo  
{  
 public static void main(String[] args)  
 {  
 File f1 = new File(args[0]);  
 File f2 = new File(args[1]);  
 f1.renameTo(f2);  
 System.*out*.println("Rename File " +f1+" To "+f2+" Sucessfully ");  
 }  
}

Task 10:

Write a code to reverse  a string..

package June21;  
  
public class ReverseStringExample {  
 public static void main(String[] args) {  
 String original = "Jaysree Hariharan";  
 String reversed = "";  
  
 // Loop through the string in reverse order  
 for (int i = original.length() - 1; i >= 0; i--) {  
 reversed += original.charAt(i);  
 }  
  
 System.*out*.println("Original: " + original);  
 System.*out*.println("Reversed: " + reversed);  
 }  
}

output:

Original: Jaysree Hariharan

Reversed: narahiraH eersyaJ

Process finished with exit code 0

Task 11:

@FunctionalInterface

interface MyInterface {

    // abstract method

    String reverse(String n);

}

public class Main {

    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"));

    }

}

Code:

package June21;  
  
  
@FunctionalInterface  
interface MyInterface11 {  
  
 // abstract method  
 String reverse(String n);  
}  
  
public class Task11\_1 {  
  
 public static void main( String[] args ) {  
  
 // declare a reference to MyInterface  
 // assign a lambda expression to the reference  
 MyInterface11 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"));  
 }  
  
}

package June21;  
  
  
@FunctionalInterface  
interface MyInterface11 {  
  
 // abstract method  
 String reverse(String n);  
}  
  
public class Task11\_1 {  
  
 public static void main( String[] args ) {  
  
 // declare a reference to MyInterface  
 // assign a lambda expression to the reference  
 MyInterface11 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

Process finished with exit code 0

Task 12:

Wap to create an arraylist with 5 friends names..

package June21;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class Task12 {  
 public static void main(String[] args) {  
 List<String> names = new ArrayList<>();  
 names.add("jaysree hariharan");  
 names.add("isha easwaran");  
 names.add("sangeetha sanjay");  
 names.add("saraswathi venugopal");  
 names.add("charan thangarajan");  
  
 names.forEach(System.*out*::println);  
  
 }  
}

output:

jaysree hariharan

isha easwaran

sangeetha sanjay

saraswathi venugopal

charan thangarajan

Process finished with exit code 0

Task 13:

Wap to create a List of 5 friends names (first name and last name)

import java.util.ArrayList;  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class Task12 {  
 public static void main(String[] args) {  
 List<String> names = new ArrayList<>();  
 names.add("jaysree hariharan");  
 names.add("isha easwaran");  
 names.add("sangeetha sanjay");  
 names.add("saraswathi venugopal");  
 names.add("charan thangarajan");  
  
 names.forEach(System.*out*::println);  
  
 }  
}

output:

jaysree hariharan

isha easwaran

sangeetha sanjay

saraswathi venugopal

charan thangarajan

Process finished with exit code 0

Task 14:

import java.util.ArrayList;

import java.util.List;

public class StreamMain {

    // 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));

    }

}

Code:

package June21;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class 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

Process finished with exit code 0

Task 15:

Collect : Terminal operator..

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

Hint:

Write a code to create a array list to store 5 integers and display the square of each no..

Hint:

List<Integer> squareofNums = numbers.stream()

w.map(num->num\*num)

.collect(Collectors.toList());

Code:

package June21;  
  
import java.util.\*;  
import java.util.stream.Collectors;  
  
public class Task15\_1 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 List<Integer> numbers = new ArrayList<>();  
  
 // Accept 5 integers from user  
 System.*out*.println("Enter 5 integers:");  
 for (int i = 0; i < 5; i++) {  
 System.*out*.print("Number " + (i + 1) + ": ");  
 numbers.add(scanner.nextInt());  
 }  
  
 // Use stream to compute squares and collect to a new list  
 List<Integer> squareOfNums = numbers.stream()  
 .map(num -> num \* num)  
 .collect(Collectors.*toList*());  
  
 // Display squared numbers  
 System.*out*.println("\nSquares of the entered numbers:");  
 for (int i = 0; i < squareOfNums.size(); i++) {  
 System.*out*.println("Square of " + numbers.get(i) + " is: " + squareOfNums.get(i));  
 }  
  
 scanner.close();  
 }  
}

output:

Enter 5 integers:

Number 1: 1

Number 2: 2

Number 3: 3

Number 4: 4

Number 5: 5

Squares of the entered numbers:

Square of 1 is: 1

Square of 2 is: 4

Square of 3 is: 9

Square of 4 is: 16

Square of 5 is: 25

Process finished with exit code 0

Task 16

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

Hint:

List<Integer> addNumbers = numbers.stream()

.filter(num -> num % 2 !=0)

.collect(Collectors.toList());

Code:

package June21;  
  
import java.util.\*;  
import java.util.stream.Collectors;  
  
public class FilterOddNumbers {  
 public static void main(String[] args) {  
 // Create an ArrayList with some numbers (you can also take input if needed)  
 List<Integer> numbers = Arrays.*asList*(10, 15, 22, 33, 41);  
  
 // Filter odd numbers using streams and collect into a new list  
 List<Integer> oddNumbers = numbers.stream()  
 .filter(num -> num % 2 != 0)  
 .collect(Collectors.*toList*());  
  
 // Display the odd numbers  
 System.*out*.println("Odd numbers are: " + oddNumbers);  
 }  
}

output:

Odd numbers are: [15, 33, 41]

Process finished with exit code 0

Task 17:

Wap to create an array list to remove duplicate values from the List.

Hint:

List<Integer> RemovDups= numbers.stream()

.distinct()

.collect(Collectors.toList());

Code:

package June21;  
import java.util.\*;  
import java.util.stream.Collectors;  
  
public class RemoveDuplicates {  
 public static void main(String[] args) {  
 // Create an ArrayList with duplicate values  
 List<Integer> numbers = Arrays.*asList*(10, 20, 10, 30, 40, 20, 50);  
  
 // Remove duplicates using streams and collect to new list  
 List<Integer> removedDups = numbers.stream()  
 .distinct()  
 .collect(Collectors.*toList*());  
  
 // Display the list after removing duplicates  
 System.*out*.println("List after removing duplicates: " + removedDups);  
 }  
}

output:

List after removing duplicates: [10, 20, 30, 40, 50]

Process finished with exit code 0

Task 18:

Wap to run a loop / iterate()  and limit it to 20 values (1 to 2)

While displaying use for each to limit till 10 numbers.

Hint:

Stream<Integers> nums = Stream

.iterate(1, n -> n+1)

.limit(20);

Nums

.limit(10)

.foreach(System.out::println);

Code:

package June21;  
  
import java.util.stream.Stream;  
  
public class Task18 {  
 public static void main(String[] args) {  
 Stream<Integer> nums = Stream.*iterate*(1, n -> n + 1)  
 .limit(20);  
  
 // Limit the stream further to 10 and print each number  
 nums.limit(10)  
 .forEach(System.*out*::println);  
 }  
}

output:

1

2

3

4

5

6

7

8

9

10

Process finished with exit code 0

Task 19:

Wap to create an array List skip 15 numbers and print the output using foreach loop

HInt:

Stream<Integers> nums = Stream

.iterate(1, n -> n+1)

.limit(20);

Stream<Integer> SkipNums = nums.skip(15);

Nums.foreach(System.out::println);

Code:

package June21;  
  
import java.util.stream.Stream;  
  
public class Task19\_1 {  
 public static void main(String[] args) {  
 Stream<Integer> nums = Stream.*iterate*(1, n -> n + 1)  
 .limit(20);  
  
 // Skip first 15 numbers  
 Stream<Integer> skipNums = nums.skip(12);  
  
 // Print the remaining numbers  
 skipNums.forEach(System.*out*::println);  
 }  
}

output:

13

14

15

16

17

18

19

20

Process finished with exit code 0