Set b 1) import java.io.\*;

public class FileCopyTransform {

public static void main(String[] args) throws IOException {

FileReader fr = new FileReader("source.txt");

FileWriter fw = new FileWriter("target.txt");

int ch;

while ((ch = fr.read()) != -1) {

char c = (char) ch;

if (Character.isDigit(c)) {

fw.write('\*');

} else if (Character.isUpperCase(c)) {

fw.write(Character.toLowerCase(c));

} else if (Character.isLowerCase(c)) {

fw.write(Character.toUpperCase(c));

} else {

fw.write(c);

}

}

fr.close();

fw.close();

System.out.println("File copied with transformations.");

}

}

2) import java.io.\*;

import java.util.Scanner;

public class AsciiWriter {

public static void main(String[] args) throws IOException {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a string: ");

String input = sc.nextLine();

FileWriter fw = new FileWriter("ascii.txt");

for (char c : input.toCharArray()) {

fw.write((int) c + "\n");

}

fw.close();

System.out.println("ASCII values written to file.");

}

}

3)

import java.util.Scanner;

class SmallNumberException extends Exception {

SmallNumberException(String msg) {

super(msg);

}

}

class LargeNumberException extends Exception {

LargeNumberException(String msg) {

super(msg);

}

}

public class FactorialCheck {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a number: ");

int num = sc.nextInt();

try {

if (num < 5) {

throw new SmallNumberException("Number is small");

} else if (num > 10) {

throw new LargeNumberException("Number is Greater");

} else {

int fact = 1;

for (int i = 1; i <= num; i++) {

fact \*= i;

}

System.out.println("Factorial: " + fact);

}

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

}  
  
4) import java.io.\*;

import java.util.\*;

public class ReverseFileContent {

public static void main(String[] args) throws IOException {

BufferedReader br = new BufferedReader(new FileReader("data.txt"));

StringBuilder sb = new StringBuilder();

String line;

while ((line = br.readLine()) != null) {

sb.append(line).append("\n");

}

br.close();

System.out.println("Reversed content:");

System.out.println(sb.reverse().toString());

}

}  
  
5) import java.io.\*;

public class ReverseWords {

public static void main(String[] args) throws IOException {

BufferedReader br = new BufferedReader(new FileReader("words.txt"));

String line;

while ((line = br.readLine()) != null) {

for (String word : line.split("\\s+")) {

System.out.println(new StringBuilder(word).reverse());

}

}

br.close();

}

}

Set c 1)

import java.io.\*;

public class DeleteTxtFiles {

public static void main(String[] args) {

for (String filename : args) {

File file = new File(filename);

if (file.exists()) {

if (filename.endsWith(".txt")) {

if (file.delete()) {

System.out.println("Deleted: " + filename);

} else {

System.out.println("Failed to delete: " + filename);

}

} else {

System.out.println("File: " + file.getName());

System.out.println("Location: " + file.getAbsolutePath());

System.out.println("Size: " + file.length() + " bytes\n");

}

} else {

System.out.println("File not found: " + filename);

}

}

}

}  
  
2) import java.io.\*;

public class ListTxtFiles {

public static void main(String[] args) {

File dir = new File("your\_directory\_path\_here"); // Replace with actual path

File[] files = dir.listFiles((d, name) -> name.endsWith(".txt"));

if (files != null) {

System.out.println(".txt files in directory:");

for (File f : files) {

System.out.println(f.getName());

}

} else {

System.out.println("Directory not found or empty.");

}

}

}  
  
3) import java.io.\*;

public class FileStats {

public static void main(String[] args) throws IOException {

BufferedReader br = new BufferedReader(new FileReader("sample.txt"));

int lines = 0, words = 0, chars = 0;

String line;

while ((line = br.readLine()) != null) {

lines++;

words += line.split("\\s+").length;

chars += line.length();

}

br.close();

System.out.println("Lines: " + lines);

System.out.println("Words: " + words);

System.out.println("Characters: " + chars);

}

}  
  
4)  
import java.io.\*;

public class CharCategory {

public static void main(String[] args) throws IOException {

FileReader fr = new FileReader("input.txt");

int ch;

while ((ch = fr.read()) != -1) {

char c = (char) ch;

if (Character.isLetter(c)) {

if (Character.isUpperCase(c)) {

System.out.print(Character.toLowerCase(c));

} else {

System.out.print(Character.toUpperCase(c));

}

} else if (Character.isDigit(c)) {

System.out.println("\nDigit: " + c);

} else if (Character.isSpaceChar(c)) {

System.out.println("\nSpace detected");

} else {

System.out.println("\nOther character: " + c);

}

}

fr.close();

}

}  
  
5) import java.util.Scanner;

// Custom Exception

class InvalidDataException extends Exception {

InvalidDataException(String message) {

super(message);

}

}

public class ValidateUserData {

// PAN format: 5 letters + 4 digits + 1 letter (e.g., ABCDE1234F)

public static boolean isValidPAN(String pan) {

return pan.matches("[A-Z]{5}[0-9]{4}[A-Z]");

}

// Mobile format: 10 digits starting with 6-9

public static boolean isValidMobile(String mobile) {

return mobile.matches("[6-9][0-9]{9}");

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter PAN number: ");

String pan = sc.nextLine().toUpperCase();

System.out.print("Enter Mobile number: ");

String mobile = sc.nextLine();

try {

if (!isValidPAN(pan) || !isValidMobile(mobile)) {

throw new InvalidDataException("Invalid Data");

}

System.out.println("PAN: " + pan);

System.out.println("Mobile: " + mobile);

} catch (InvalidDataException e) {

System.out.println("Error: " + e.getMessage());

}

}

}