# IO Streams

## [1. Byte stream](https://personales.unican.es/corcuerp/java/Labs/LAB_20.htm#Exercise_1)

import java.io.\*;  
  
public class FileInputOutputStream {  
     
    public static void main(String[] args) throws IOException {  
         
        File inputFile = new File("InputFile.txt");  
        File outputFile = new File("OutputFile.txt");  
         
      **FileInputStream in = new FileInputStream(inputFile);  
        FileOutputStream out = new FileOutputStream(outputFile);**  
        int c;  
         
        while ((c = in.read()) != -1){  
            System.out.println(c);  
            out.write(c);  
        }  
         
        System.out.println("FileInputStream is used to read a file and FileOutPutStream is used for writing.");  
         
        in.close();  
        out.close();  
    }  
}

## [2. Character stream](https://personales.unican.es/corcuerp/java/Labs/LAB_20.htm#Exercise_2)

import java.io.\*;  
  
public class FileReaderWriter {  
     
    public static void main(String[] args) throws IOException {  
         
        File inputFile = new File("InputFile.txt");  
        File outputFile = new File("OutputFile.txt");  
         
        **FileReader in = new FileReader(inputFile);  
        FileWriter out = new FileWriter(outputFile);**  
        int c;  
         
        while ((c = in.read()) != -1){  
            System.out.println(c);  
            out.write(c);  
        }  
         
        System.out.println("FileReader is used to read a file and FileWriter is used for writing.");  
         
        in.close();  
        out.close();  
    }  
}

## [3. Buffered stream](https://personales.unican.es/corcuerp/java/Labs/LAB_20.htm#Exercise_3)

public class BufferedReaderWriter {  
     
    public static void main(String args[]) {  
         
        String a0, a1, a2;   
            a0 = "words.txt";  
            a1 = "wordsout.txt";  
            a2 = "3"; // dịch đi 3 vị trí  
          
    **SimpleEncryption se = new SimpleEncryption();**

**se.encrypt(a0, a1, Integer.parseInt(a2));// Mã hóa file**  
         
         System.out.println("Hiển thị file đã bị mã hóa");  
        **se.viewFileContent(a1);**  
         
    }  
}

class SimpleEncryption {  
     
   void **encrypt**(String source, String dest, int shiftSize) {  
        
      BufferedReader reader;  
      BufferedWriter writer;  
       
      try {  
 **reader = new BufferedReader(new FileReader(source));  
         writer = new BufferedWriter(new FileWriter(dest));**  
         String line = reader.readLine();  
         int data;  
         while (line != null) {  
            for (int i = 0; i < line.length(); i++) {  
               data =(int) line.charAt(i) + shiftSize;  
               writer.write(data);  
            }  
            writer.write((int)'\n');  
            line = reader.readLine();  
         }  
         reader.close();  
         writer.close();  
      } catch (IOException ie) {  
         System.out.println("Failed encrypting the file content.");  
      }  
  
   }  
    
   void **viewFileContent**(String filename) {  
        
      **BufferedReader reader;**  
      try {  
         **reader = new BufferedReader(new FileReader(filename));**  
         String line = reader.readLine();  
         while (line != null) {  
            System.out.println(line);  
            line = reader.readLine();  
         }  
         reader.close();  
      } catch (IOException ie) {  
         System.out.println("Failed to open file for reading.");  
      }  
   }  
    
}

## [4. Data stream](https://personales.unican.es/corcuerp/java/Labs/LAB_20.htm#Exercise_4)

import java.io.\*;  
  
public class DataInputOutputStream {  
     
    public static void main(String[] args) throws IOException {         
        // write the data out  
        **DataOutputStream out = new DataOutputStream(new  
                FileOutputStream("invoice.dat"));**  
         
        double[] prices = { 19.99, 9.99, 15.99, 3.99, 4.99 };  
        int[] units = { 12, 8, 13, 29, 50 };  
        String[] descs = { "Java 1",  
        "Java 2",   
        "Java SE",  
        "Java EE",  
        "Java Key Chain" };  
         
        for (int i = 0; i < prices.length; i ++) {  
      **out.writeDouble(prices[i]);  
            out.writeChar('\t');  
            out.writeInt(units[i]);  
            out.writeChar('\t');  
            out.writeChars(descs[i]);  
            out.writeChar('\n');**  
        }  
        out.close();  
         
        // read it in again  
        **DataInputStream in = new DataInputStream(new  
                FileInputStream("invoice.dat"));**  
         
        double price;  
        int unit;  
        StringBuffer desc;  
        double total = 0.0;  
         
        String lineSepString = System.getProperty("line.separator");  
        char lineSep = lineSepString.charAt(lineSepString.length()-1);  
         
        try {  
            while (true) {  
**price = in.readDouble();  
                in.readChar();       // throws out the tab  
                unit = in.readInt();  
                in.readChar();       // throws out the tab  
                char chr;  
                desc = new StringBuffer(20);  
                while ((chr = in.readChar()) != lineSep)  
                    desc.append(chr);  
                System.out.println("You've ordered " +  
                        unit + " units of " +  
                        desc + " at $" + price);**  
                total = total + unit \* price;  
            }  
        } catch (EOFException e) {  
        }  
        System.out.println("For a TOTAL of: $" + total);  
        in.close();  
    }  
}

## [5. Object stream](https://personales.unican.es/corcuerp/java/Labs/LAB_20.htm#Exercise_5)

import java.io.\*;  
  
public class ObjectInputOutputStream {  
     
    public static void main(String[] args) {  
         
        Card3 card = new Card3(12, Card3.SPADES);  
        System.out.println("Card to write is: " + card);  
         
        try {  
           **FileOutputStream out = new FileOutputStream("card.out");  
            ObjectOutputStream oos = new ObjectOutputStream(out);  
            oos.writeObject(card);  
            oos.flush();**  
        } catch (Exception e) {  
            System.out.println("Problem serializing: " + e);  
        }  
         
        try {  
  **FileInputStream in = new FileInputStream("card.out");  
            ObjectInputStream ois = new ObjectInputStream(in);  
            card = (Card3)(ois.readObject());**  
        } catch (Exception e) {  
            System.out.println("Problem serializing: " + e);  
        }  
         
        System.out.println("Card read is: " + card);  
    }  
}

## 6. Bài tập

Đọc dữ liệu từ file DanhSach.txt

Săp xếp theo tên\_ho\_đệm và ghi ra file DSOutPut.txt

(Ko được dùng các cấu trúc List,…)

import java.io.Serializable;  
  
**public class Card3 implements Serializable** {  
    private int suit = UNASSIGNED;  
    private int number = UNASSIGNED;  
  
    public final static int UNASSIGNED = -1;  
  
    public final static int DIAMONDS = 1;  
    public final static int CLUBS = 2;  
    public final static int HEARTS = 3;  
    public final static int SPADES = 4;  
  
    public final static int ACE = 1;  
    public final static int KING = 13;  
  
    public Card3(int number, int suit) {  
        if (isValidNumber(number)) {  
            this.number = number;  
        } else {  
            //Error  
        }  
  
        if (isValidSuit(suit)) {  
            this.suit = suit;  
        } else {  
            //Error  
        }  
    }  
  
    public int getSuit() {  
        return suit;  
    }  
  
    public int getNumber() {  
        return number;  
    }  
  
    public static boolean isValidNumber(int number) {  
        if (number >= ACE && number <= KING) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
    public static boolean isValidSuit(int suit) {  
        if (suit >= DIAMONDS && suit <= SPADES) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
  
    public boolean equals(Object obj) {  
        if (obj instanceof Card3) {  
        Card3 card = (Card3)obj;  
            if (card.getNumber() == this.number && card.getSuit() == this.suit) {  
                return true;  
            } else {  
                return false;  
            }  
        } else {  
            return false;  
        }  
    }  
    public int hashCode() {  
        return number \* suit;  
    }  
    public String toString() {  
        return numberToString(this.number) + " of "  
               + suitToString(this.suit);  
    }  
  
    public static String numberToString(int number) {  
        String result = "";  
        switch (number) {  
            case ACE: result =  "Ace"; break;  
            case 2: result = "Two"; break;  
            case 3: result = "Three"; break;  
            case 4: result = "Four"; break;  
            case 5: result = "Five"; break;  
            case 6: result = "Six"; break;  
            case 7: result = "Seven"; break;  
            case 8: result = "Eight"; break;  
            case 9: result = "Nine"; break;  
            case 10: result = "Ten"; break;  
            case 11: result = "Jack"; break;  
            case 12: result = "Queen"; break;  
            case KING: result = "King"; break;  
            case UNASSIGNED: result = "Invalid Number"; break;  
        }  
        return result;  
    }  
  
    public static String suitToString(int suit) {  
        String result = "";  
        switch (suit) {  
            case DIAMONDS: result = "Diamonds"; break;  
            case CLUBS: result = "Clubs"; break;  
            case HEARTS: result = "Hearts"; break;  
            case SPADES: result = "Spades"; break;  
            case UNASSIGNED: result = "Invalid Suit"; break;  
        }  
        return result;  
    }  
}

## 