Input Output

1. Define stream in Java. Describe how stream is used in both input and output.

Or

What is stream? How is the concept of stream used in Java?

Ans:

Stream:

A stream in Java is a path along which data flows. It has a source (of data) and a destination (for that data) as depicted in below figure. Both the source and the destination may be physical devices or programs or other streams in the same program.

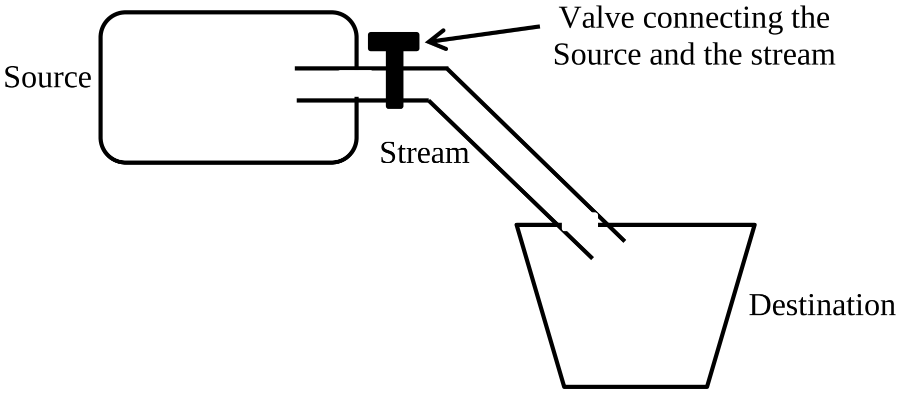
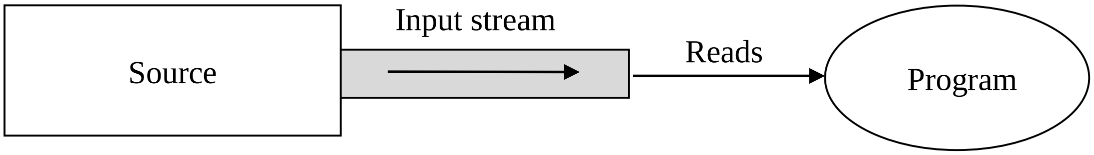
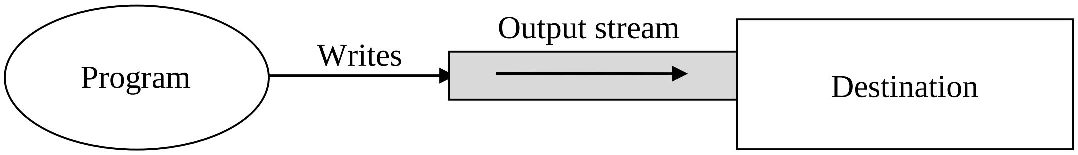


Fig: Conceptual view of a stream

Java streams are classified into two basic types, namely, *input stream* and *output stream*. An input stream extracts (i.e reads) data from the source (file) and sends it to the program. Similarly, an output stream takes data from the program and sends (i.e writes) it to the destination (file). The given figure illustrates the use of input and output streams:



a) Reading data into a program.



b) Writing data to a destination

2. Briefly describe the hierarchy of java stream classes.

Or

Briefly describe the stream classes which are contained in java.io package.

Ans:

The `java.io` package contains a large number of stream classes that provide capabilities for processing all types of data. These classes may be categorized into two groups based on the data type on which they operate.

1. Byte stream classes that provide support for handling I/O operations on bytes.
2. Character stream classes that provide support for managing I/O operations on characters.

These two groups may further be classified based on their purpose. The figure shows how stream classes are grouped based on their functions.

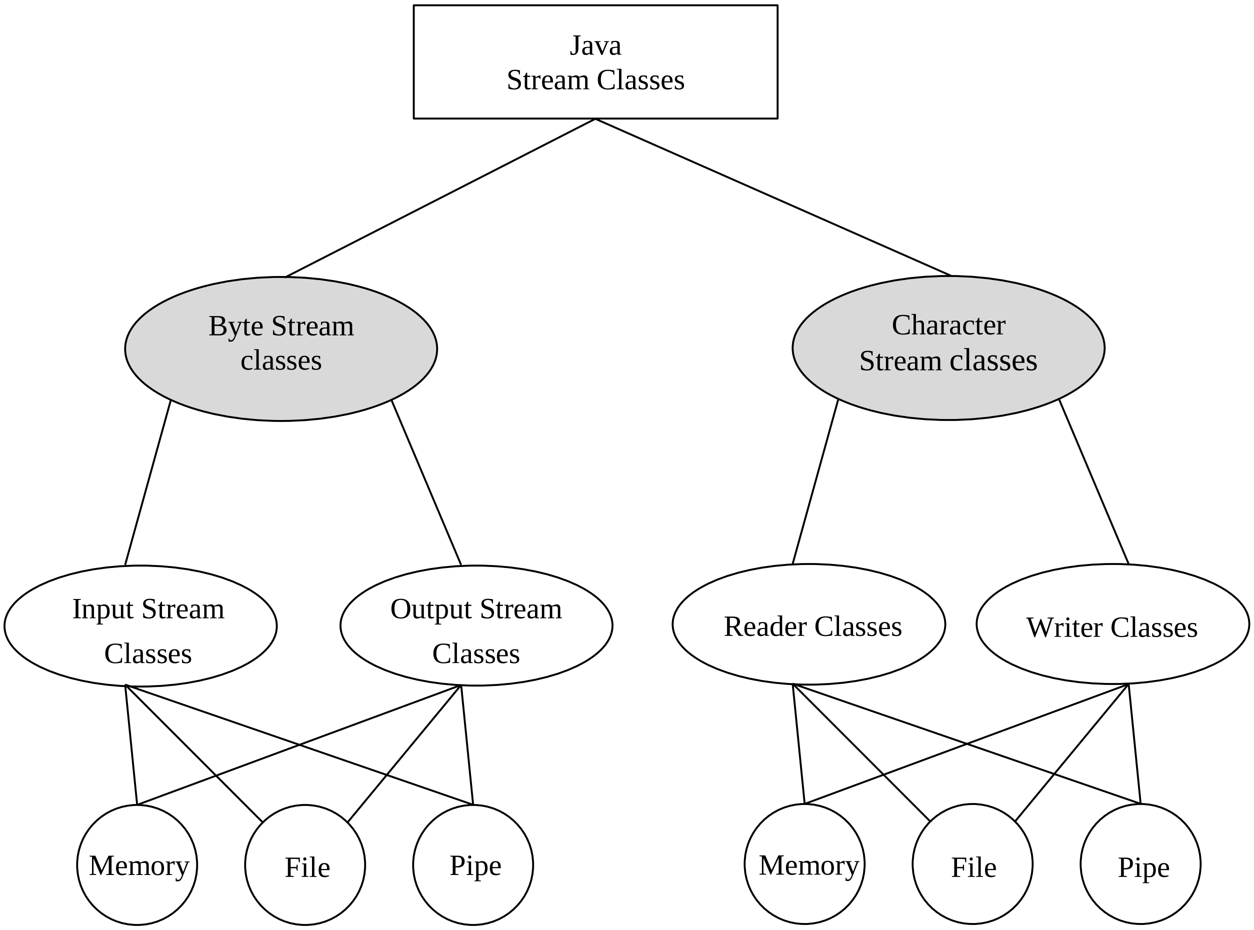


Fig: Classification of Java stream classes

3. Show the hierarchy of Input Stream Classes.
Ans:

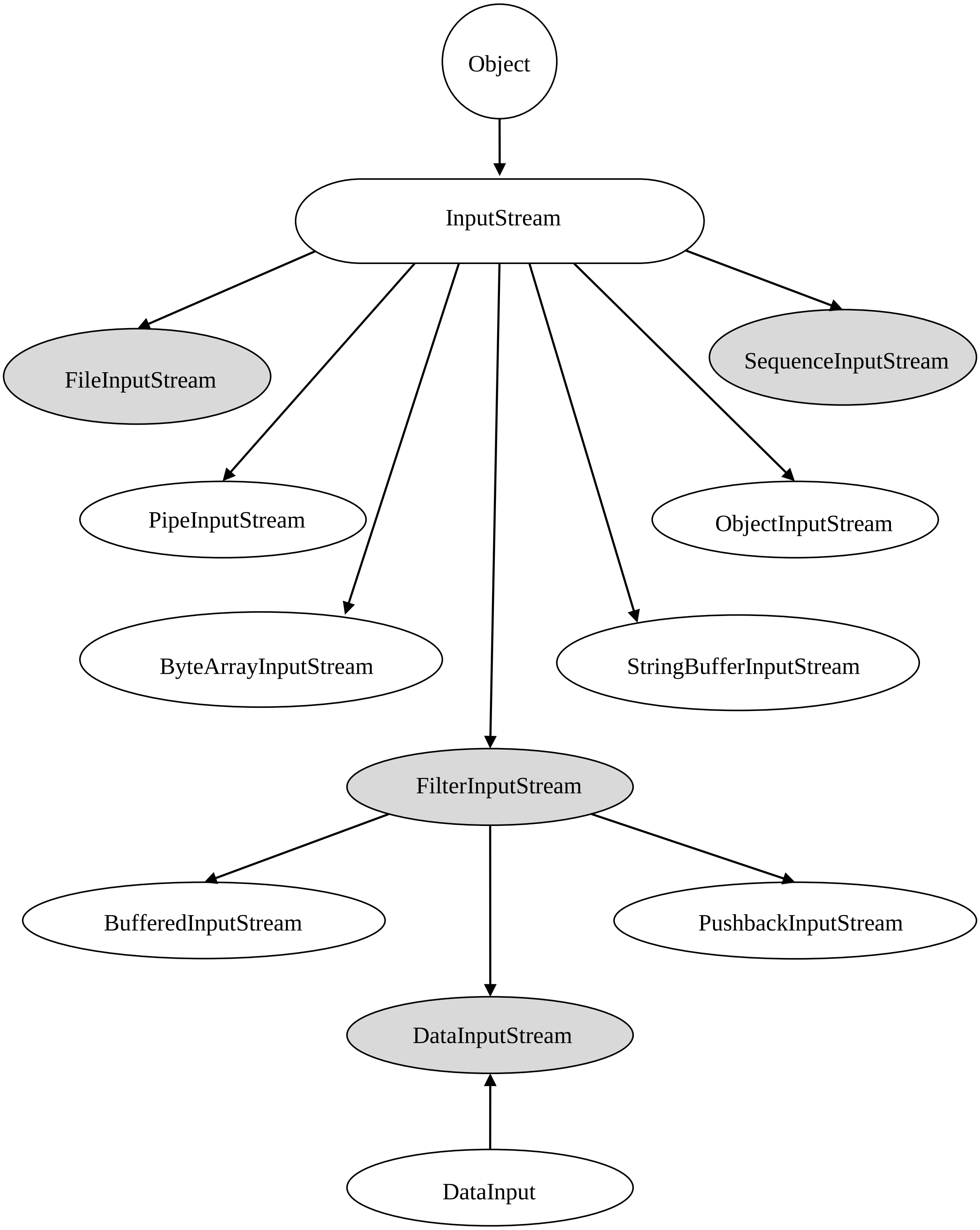


Fig: Hierarchy of input stream classes

4. Show the hierarchy of Output Stream Classes.
Ans:

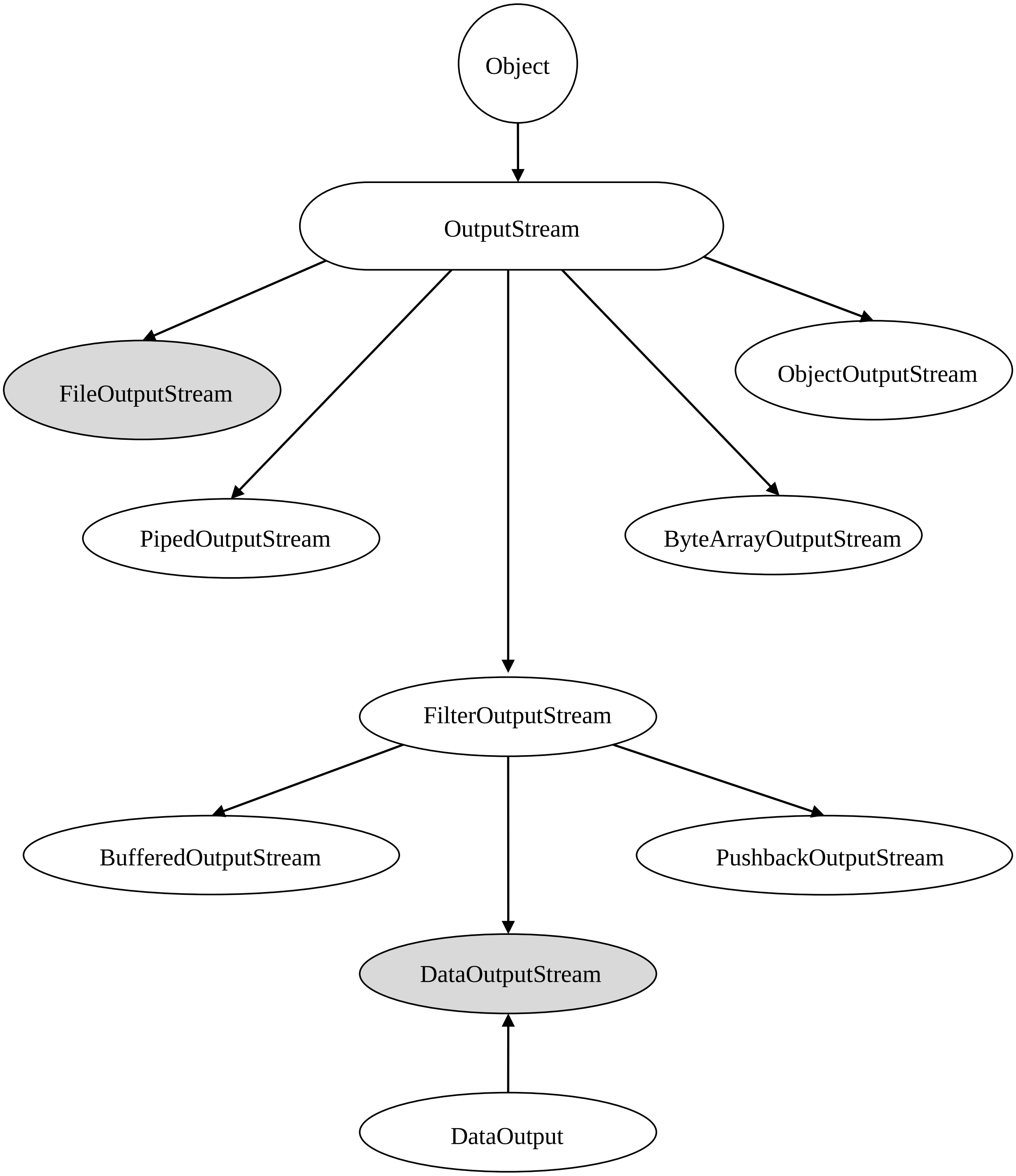


Fig: Hierarchy of output stream classes

5. Write the purposes of the following functions and the name of the classes they belongs to: i) writeBytes( ) ii) write( ) iii) close( ) iv) read( ) v) writeInt ( ) vi) readLine( ) vii) flush( )

Ans:

Functions	**Purposes**	**Classes belongs to**
`read( )`	Reads a byte from the input stream	InputStream
`read(byte b[ ])`	Reads an array of bytes into b	InputStream
`available()`	Gives number of bytes available for input	InputStream
`skip(n)`	Skips over n bytes from the input stream	InputStream
`reset( )`	Goes back to the beginning of the stream	InputStream
`close( )`	Closes the input stream	InputStream
`readLine( )`	Reads the next line of text from the input stream.	DataInputStream
`readInt( )`	Reads four input bytes and returns an intvalue.	DataInputStream
`readUTF( )`	Reads in a string that has been encoded using a modified UTF-8 format.	DataInputStream
<u>Found details on the following link</u> http://pic.dhe.ibm.com/infocenter/adiehelp/v5r1m1/topic/com.sun.api.doc/java/io/DataInput.html & Table-16.1 (page 287) of Programming with Java-Balagurusamy		
`write( )`	Writes a byte to the output stream	OutputStream
`write(byte[] b)`	Write all bytes in the array b to the output stream	OutputStream
`flush( )`	Flushes the output stream	OutputStream
`writeInt ( )`	Writes an int value, which is comprised of four bytes, to the output stream.	DataOutputStream
`writeBytes( )`	Writes a string to the output stream	DataOutputStream
`writeBoolean( )`	Writes a boolean value to this output stream.	DataOutputStream
`writeChar( )`	Writes a char value, which is comprised of two bytes, to the output stream.	DataOutputStream
<u>Found details on the following link</u> http://pic.dhe.ibm.com/infocenter/adiehelp/v5r1m1/topic/com.sun.api.doc/java/io/DataOutput.html & Table-16.2 (page 289) of Programming with Java-Balagurusamy		

6. What are the functions of DataInputStream and DataOutputStream classes? How do they work? Explain with some illustration.

Ans:

DataInputStream:

DataInputStream reads bytes from the stream and converts them into appropriate primitive type values or strings. DataInputStream extends FilterInputStream and implements the DataInput interface. DataInputStream implements the methods defined in the DataInput interface to read primitive data type values and strings. DataInputStream defines the following constructor:

```
DataInputStream(InputStream inputStream);
```

DataOutputStream:

DataOutputStream converts primitive type values or strings into bytes and outputs the bytes to the stream. DataOutputStream extends FilterOutputStream and implements the DataOutput interface. DataOutputStream implements the methods defined in the DataOutput interface to write primitive data type values and strings.

```
DataOutputStream(OutputStream outputStream);
```

The following program demonstrates the use of DataInputStream and DataOutputStream classes:

```
import java.io.*;

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

      FileOutputStream fout=new FileOutputStream("D:/Test.txt");
      DataOutputStream out=new DataOutputStream(fout);

      out.writeDouble(98.6);
      out.writeInt(1000);
      out.writeBoolean(true);
      out.close();

      FileInputStream fin=new FileInputStream("D:/Test.txt");
      DataInputStream in=new DataInputStream(fin);

      double d=in.readDouble();
      int i=in.readInt();
      boolean b=in.readBoolean();

      System.out.println("Here are the values: "+d+" "+i+" "+b);
      in.close();

     }
}
```

7. What is the function of Stream Tokenizer.
Ans:

The class Stream Tokenizer, a subclass of object can be used for breaking up a stream of text from an input text file into meaningful pieces called tokens.

8. What is a file? Why do we require files to store data?
Ans:

File:
A file is a collection of related records placed in a particular area on the disk. A record is composed of several fields and a field is a group of characters. Characters in Java are Unicode characters composed of two bytes and each byte containing eight binary digits 1 or 0.

Inside the programs, storing data by using variables and arrays poses the following problem:

1. The data is lost either when a variable goes out of scope or when the program is terminated. That is the storage is temporary.
2. It is difficult to handle large volumes of data using variables and arrays.

We can overcome this problem by storing data secondary storage devices such as floppy disks or hard disks. The data is stored in these devices using the concept of files. Data stored in files is often called persistent data. That's why we require files to store data.

9. What is a random access file? How is it different from a sequential access file?
Ans:

Random Access file:
The `RandomAccessFile` class supported by the java.io package allows us to create files that can be used for reading and writing data with random access. That is, we can "jump around" in the file while using the file. Such files are known as random access file.

Difference between random and sequential access file:
Sequential access file can be used either for "read only" or for "write only" operations and not for both purposes simultaneously. Whereas in random access file we can "jump around" the file while using the file that is reading and writing operations can be performed simultaneously.

10. Specify the two mode strings used to handle Random Access File with their purpose.
Ans:

A file can be created and opened for random access by giving a mode string as a parameter to the constructor when we open the file. We can use one of the following two mode strings:

- "`r`" for reading only
- "`rw`" for both reading and writing

An existing file can be updated using the "`rw`" mode.

11. Write a program of copy the contents of a file named “INPUT.DAT” into a file called “OUPUT.DAT”.

Or

Write a program that reads characters from a file “in.txt” and writes them into another file “out.txt” using character stream.

Ans:

```
import java.io.*;

public class CopyCharacters {
     public static void main(String[] args) {
          File inFile = new File("D:/input.dat");
          File outFile = new File("D:/output.dat");

          FileReader ins = null;
          FileWriter outs = null;


          try{
               ins=new FileReader(inFile);
               outs = new FileWriter(outFile);

               int ch;
               while((ch=ins.read()) !=-1){
                    outs.write(ch);
               }
          }

          catch(IOException e){
               System.out.println(e);
               System.exit(-1);
          }

          finally{
               try{
                    ins.close();
                    outs.close();
               }
               catch(IOException e){
                    System.out.println(e);
               }
          }
     }
}
```

12. Write a program to copy bytes from “file_a” to “file_b”. Assume that the files are stored in C:\ directory. If you need to import any class/package then, show it in your program.

Ans:

```
import java.io.*;

public class CopyBytes {

    public static void main(String[] args) {

        FileInputStream infile = null;
        FileOutputStream outfile = null;

        byte byteRead;

        try {
            infile = new FileInputStream("D:/file_a.dat");
            outfile = new FileOutputStream("D:/file_b.dat");

            do {
                byteRead = (byte) infile.read();
                outfile.write(byteRead);
            }
            while (byteRead != -1);
        }

        catch (FileNotFoundException e) {
            System.out.println("File not found");
        }
        catch (IOException e) {
            System.out.println(e.getMessage());

        }

        finally
        {
            try {
                infile.close();
                outfile.close();
            }
            catch (IOException e) {
                System.out.println(e);
            }
        }
    }
}
```

13. Write a program to read several numbers from a file, then write the even numbers of them to “event.txt” and odd numbers to “odd.txt”.

Ans:

```
import java.io.*;
import java.io.IOException;

public class OddEven {
      public static void main(String[] args) {
            File inFile = new File("D:/num.txt");
            File oddFile = new File("D:/odd.txt");
            File evenFile = new File("D:/even.txt");

            BufferedReader ins = null;
            FileWriter odd = null;
            FileWriter even = null;

            try{
                  ins = new BufferedReader(new FileReader(inFile));
                  odd = new FileWriter(oddFile);
                  even = new FileWriter(evenFile);

            String intr="";
            String filedata="";
                  while((intr=ins.readLine()) !=null){
                        filedata+=intr+" ";
                  }

                  String delims = " ";
                  String[] token=filedata.split(delims);
                  int word;
                  for(int i=0;i<token.length;i++){
                          word = Integer.parseInt( token[i]);
                          if(word%2==0)
                              even.write(word+" ");
                          else
                              odd.write(word+"  ");
                        }
            }

            catch(IOException e){
                  System.out.println(e);
                  System.exit(-1);
            }

            finally{
                  try{
                        ins.close();
                        odd.close();
                        even.close();
                  }
                  catch(IOException e){
                        System.out.println(e);
                  }
            }
      }
}
```

14. **Write a program to copy bytes from “file_a” to “file_b”. Assume that the files are stored in C:\ directory. If you need to import any class/package then, show it in your program.**

Ans:

```
import java.io.*;

public class CopyBytes {
     public static void main(String[] args) {

          FileInputStream infile = null;
          FileOutputStream outfile = null;

          byte byteRead;

          try {
               infile = new FileInputStream("D:/file_a.txt");
               outfile = new FileOutputStream("D:/file_b.txt");

               do {
                    byteRead = (byte) infile.read();
                    outfile.write(byteRead);
               }
               while (byteRead != -1);
          }

          catch (FileNotFoundException e) {
               System.out.println("File not found");
          }

          catch (IOException e) {
               System.out.println(e.getMessage());
          }

          finally{
               try{
                    infile.close();
                    outfile.close();
               }
               catch (IOException e) {
                    System.out.println(e);
               }
          }
     }
}
```

15. Write a skeleton Java program to read a line of text from the keyboard and write it in a file called “out.text”.

Ans:

```
import java.io.*;
import java.util.Scanner;

public class ReadLine {
     public static void main(String[] args) {

          File outFile = new File("D:/output.dat");

          FileWriter outs = null;

          try{
          outs = new FileWriter(outFile);

          Scanner scan=new Scanner(System.in);
          String str=scan.nextLine();
          outs.write(str);

          }

          catch(IOException e){
               System.out.println(e);
               System.exit(-1);
          }

          finally{
               try{
                    outs.close();
               }
               catch(IOException e){
                    System.out.println(e);
               }
          }
     }
}
```

16. Write a Java program that will count the number of character in a file.
Ans:

```
import java.io.*;

public class CopyCharacters {
    public static void main(String[] args) {

        File inFile = new File("D:/output.dat");
        FileReader ins = null;

        try {
            ins = new FileReader(inFile);

            int ch;
            int i = 0;
            while ((ch = ins.read()) != -1) {
                if (ch != ' ')
                    i++;
            }
            System.out.println("Number of Character: "+i);
        }

        catch (IOException e) {
            System.out.println(e);
            System.exit(-1);
        }

        finally {
            try {
                ins.close();
            } catch (IOException e) {
                System.out.println(e);
            }
        }
    }
}
```

17. Write a Java program that will count the number of words in a file.
Ans:

```
import java.io.*;

public class WordCount {
     public static void main(String[] args) {

          File inFile = new File("D:/output.dat");
          BufferedReader ins = null;

          try{
               ins = new BufferedReader(new FileReader(inFile));

               String intr="";
               String filedata="";
               while((intr=ins.readLine()) !=null){
                    filedata+=intr+" ";
               }

               String delims = " ";
               String[] token=filedata.split(delims);
               System.out.print("Word No.: "+token.length+"\n");
          }

          catch(IOException e){
               System.out.println(e);
               System.exit(-1);
          }

          finally{
               try{
                    ins.close();
               }
               catch(IOException e){
                    System.out.println(e);
               }
          }
     }
}
```

18. Write a program to write 100 randomly generated integer to a file called “rand.txt” and then read the contents of the file and display them on the screen in the sorted manner.
Ans:

```
import java.io.*;

public class WriteReadIntegers {
     public static void main(String[] args) {
          DataInputStream dis =null;
          DataOutputStream dos =null;
          File intFile=new File ("D:/rand.txt");

          try{

     dos=new DataOutputStream(new FileOutputStream(intFile));

               for(int i=0;i<20;i++){
                    dos.writeInt ((int)(Math.random( )*100)) ;
               }
               dos.close();
          }

       catch(IOException ioe) {
        System.out.println(ioe.getMessage( ));
       }

       try{

        dis=new DataInputStream (new FileInputStream (intFile));

        for(int i=0;i<20;i++){
             int n=dis.readInt ();
             System.out.print(n+ " ");
        }

             dis.close();
       }

       catch(IOException ioe){
        System.out.println(ioe.getMessage( ));
       }
    }
}
```

19. Write a program that will store a Character, a Integers and a Double value in a file. Then from the same program retrieve Integer value and display the values on the standard output device.

Ans:

```
import java.io.*;

public class RandomIO {
     public static void main(String[] args) {
          RandomAccessFile file = null;

          try {
               file = new RandomAccessFile("D:/rand.dat", "rw");

               file.writeChar('X');
               file.writeInt(555);
               file.writeDouble(3.1412);

               file.seek(2);
               System.out.println(file.readInt());

               file.close();
           }

          catch (IOException e) {
               System.out.println(e);
          }
     }
}
```

20. Write a program that reads a file containing integers and appends its end the sum of all the integers.

Ans:

```
import java.io.*;

public class Append {
     public static void main(String[] args) {
          RandomAccessFile rfile;
          try  {
               rfile=new RandomAccessFile("D:/city.txt","rw");
               rfile.seek(rfile.length());
               rfile.writeBytes("CTG");
               rfile.close();
          }
          catch(IOException ioe){
               System.out.println(ioe);
          }
     }
}
```