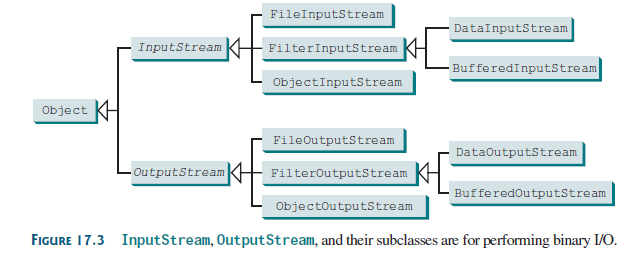
Binary I/O Classes

The abstract InputStream is the root class for reading binary data, and the abstract

OutputStream is the root class for writing binary data.

The design of the Java I/O classes is a good example of applying inheritance, where common operations are generalized in superclasses, and subclasses provide specialized operations.

Figure 17.3 lists some of the classes for performing binary I/O. InputStream is the root for



**Listing 17.1 TestFileStream.java**

import java.io.\*;

public class TestFileStream {

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

try (

// Create an output stream to the file

FileOutputStream output = new FileOutputStream("temp.dat");

) {

// Output values to the file

for (int i = 1; i <= 10; i++)

output.write(i);

}

try (

// Create an input stream for the file

FileInputStream input = new FileInputStream("temp.dat");

) {

// Read values from the file

int value;

while ((value = input.read()) != -1)

System.out.print(value + " ");

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Listing 17.2 TestDataStream.java**

public class C1\_FileStream {

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

File file = new File("src/COSC603\_2022\_03\_16\_IO/temp.dat");

try(FileInputStream input = new FileInputStream(file)){ // TRY WITH RESOURCES // Create an output stream to the file

int value;

while ((value = input.read()) != -1)

System.out.print(value + " ");

}

try( FileOutputStream output = new FileOutputStream(file, false)){

// write in Binary File

int value;

for (int i = 100; i <= 120; i++)

output.write(i);

output.close();

FileInputStream input2 = new FileInputStream(file);

while ((value = input2.read()) != -1)

System.out.print(value + " ");

input2.close();

}

package COSC603\_2022\_03\_16\_IO;

//Listing 17.2 TestDataStream.java

import java.io.\*;

public class C2\_DataStream {

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

File file = new File("src/COSC603\_2022\_03\_16\_IO/temp.dat");

try (DataOutputStream output = new DataOutputStream(new FileOutputStream(file,false))) {

output.writeUTF("Dave");

output.writeDouble(85.5);

output.writeUTF("Susan");

output.writeDouble(95.5);

output.writeUTF("Joe");

output.writeDouble(80.5);

}

// IN TRY CLAUSE ( DataOutputStream output = new DataOutputStream(new FileOutputStream(file)) )IS EQUAL two lines:

// FileOutputStream output = new FileOutputStream(file);

// DataOutputStream outputStream= new DataOutputStream(output);

try (DataInputStream input = new DataInputStream(new FileInputStream(file))) {

// Read student test scores from the file

while(true){

System.out.println(input.readUTF() + " " + input.readDouble());

}

// System.out.println(input.readUTF() + " " + input.readDouble());

// System.out.println(input.readUTF() + " " + input.readDouble());

}catch(EOFException *ex*){

System.out.println("Done reading the file");

}

// IN TRY CLAUSE (DataInputStream input = new DataInputStream(new FileInputStream(file) ) IS EQUAL two lines:

// FileInputStream input= new FileInputStream(file);

// DataInputStream inputData = new DataInputStream(input);

}

}

**17.4.4 BufferedInputStream/BufferedOutputStream**

**package COSC603\_2022\_03\_16\_IO;**

**// ADD BUFFER INPUT AND OUTPUT STREAM IN STRING**

**import java.io.\*;**

**public class C3\_BufferedStream {**

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

**File file = new File("src/COSC603\_2022\_03\_16\_IO/temp.dat");**

**try (DataOutputStream output = new DataOutputStream( new BufferedOutputStream ( new FileOutputStream(file,false) ))) {**

**output.writeUTF("Dave");**

**output.writeDouble(85.5);**

**output.writeUTF("Susan");**

**output.writeDouble(95.5);**

**output.writeUTF("Joe");**

**output.writeDouble(80.5);**

**}**

**// IN TRY CLAUSE (DataOutputStream output = new DataOutputStream( new BufferedOutputStream ( new FileOutputStream(file,false) ))) IS EQUAL two lines:**

**// FileOutputStream output = new FileOutputStream(file);**

**// ADD-> BufferedOutputStream buffOut = new BufferedOutputStream(output, 1024); // we can change default buffer size form 512Byte to any amount**

**// DataOutputStream outputStream= new DataOutputStream(buffOut);**

**// 512 bytes default buffer size**

**try (DataInputStream input = new DataInputStream( new BufferedInputStream( new FileInputStream(file) ))) {**

**// Read student test scores from the file**

**while(true){**

**System.out.println(input.readUTF() + " " + input.readDouble());**

**}**

**}catch(EOFException *ex*){**

**System.out.println("Done reading the file");**

**}**

**// IN TRY CLAUSE (DataInputStream input = new DataInputStream( new BufferedInputStream( new FileInputStream(file) ))) IS EQUAL two lines:**

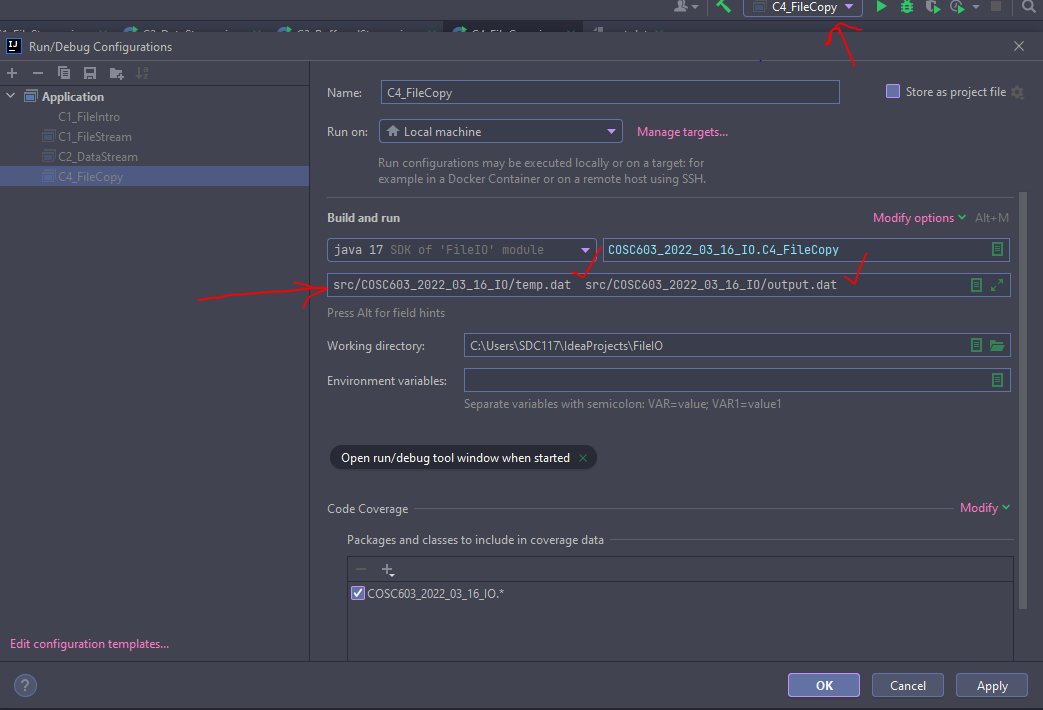
**// FileInputStream FileIN= new FileInputStream(file);**

**// ADD-> BufferedInputStream buffInput = new BufferedInputStream(FileIN)**

**// DataInputStream input = new DataInputStream(buffInput);**

**}**

**}**

****

**package COSC603\_2022\_03\_16\_IO;**

**import java.io.\*;**

**//Listing 17.4 Copy.java**

**public class C4\_FileCopy {**

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

**if (*args*.length != 2) {**

**System.out.println( "Need to enter both source and destination file name");**

**System.exit(1);**

**}**

**File source = new File(*args*[0]);**

**if (!source.exists()) {**

**System.out.println("Source file " + *args*[0] + " does not exist");**

**System.exit(2);**

**}**

**File destination = new File(*args*[1]);**

**try (**

**BufferedInputStream input = new BufferedInputStream(new FileInputStream(source));**

**BufferedOutputStream output = new BufferedOutputStream(new FileOutputStream(destination));**

**) {**

**int r, numberOfBytesCopied = 0;**

**while ((r = input.read()) != -1) {**

**output.write((byte)r);**

**numberOfBytesCopied++;**

**}**

**// Display the file size**

**System.out.println("wrote " + numberOfBytesCopied + " bytes of data.");**

**}**

**}**

**}**