Due to the print book page limit, we cannot inlcude all good CheckPoint questions in the physical book. The CheckPoint on this Website may contain extra questions not printed in the book. The questions in some sections may have been reordered as a result. Nevertheless, it is easy to find the CheckPoint questions in the book on this Website. Please send suggestions and errata to Dr. Liang at y.daniel.liang@gmail.com. Indicate the book, edition, and question number in your email. Thanks!

Section 17.2

▼17.2.1

What is a text file and what is a binary file? Can you view a text file or a binary file using a text editor?

Although it is not technically precise, a text file consists of a sequence of characters and a binary file consists of a sequence of bits. You can use a text editor to view a text file, but not a binary file.

Hide Answer

▼17.2.2

How do you read or write text data in Java? What is a stream?

You have to use Java I/O classes to create objects and use the methods in the objects to perform I/O. A Java I/O object is called a stream. An object for reading data is called an input stream and an object for writing data is called an output stream.

Hide Answer

Section 17.3

▼17.3.1

What are the differences between text I/O and binary I/O?

Binary I/O reads a byte from a file and copies it directly to the memory with any conversion, vice versa. Text I/O requires encoding and decoding. The JVM converts a Unicode to a file specific encoding when writing a character and coverts a file specific encoding to a Unicode when reading a character.

Hide Answer

▼17.3.2

How is a Java character represented in the memory, and how is a character represented in a text file?

Characters are represented using Unicode in the memory and characters are represented in a file using a specified encoding scheme. If no encoding scheme is specified, the system's default encoding scheme is used.

Hide Answer

▼17.3.3

If you write the string "ABC" to an ASCII text file, what values are stored in the file?

The values stored in the text file are 0x41 0x42 0x43.

Hide Answer

▼17.3.4

If you write the string "100" to an ASCII text file, what values are stored in the file? If you write a numeric byte-type value 100 using binary I/O, what values are stored in the file?

If you write string "100" to an ASCII text file, the values stored are 0x31 0x30 0x30. If you write a numeric byte-type value 100 using binary I/O, the value stored in the file is 0x64.

Hide Answer

▼17.3.5

What is the encoding scheme for representing a character in a Java program? By default, what is the encoding scheme for a text file on Windows?

The encoding scheme for representing a character in a Java program is the Unicode. By default, a text file is encoded using ASCII.

Hide Answer

Section 17.4

▼17.4.1

Why do you have to declare to throw IOException in the method or use a try-catch block to handle IOException for Java I/O programs?

Almost all the methods and constructors in Java I/O classes, because there are always some unexpected situation may arise during I/O.

Hide Answer

▼17.4.2

Why should you always close streams? How do you close streams?

Two reasons: (1) closing a stream ensures that data will be written to the file. (2) closing a stream releases resource acquired by the stream object.

Two ways to close a stream: 1. Invoking the close() method. 2. Using the try-catch-resource to automatically close the stream.

Hide Answer

▼17.4.3

The read() method in InputStream reads a byte. Why does it return an int instead of a byte? Find the abstract methods in InputStream and OutputStream.

The value of a byte is returned as an int in the range 0 to 255. If no byte is available because the end of the stream has been reached, the value -1 is returned. The only abstract method in InputStream is read() and the only abstract method in OutputStream is write(int).

Hide Answer

▼17.4.4
Does FileInputStream/FileOutputStream introduce any new methods beyond the methods inherited from InputStream/OutputStream? How do you create a FileInputStream/FileOutputStream?
All the methods in FileInputStream/FileOutputStream are inherited from InputStream/OutputStream. Use new FileInputStream(filename) or new FileInputStream(File) to create a new FileInputStream and use new FileOutputStream(filename), new FileOutputStream(File), new FileOutputStream(filename, true) or new FileOutputStream(File, true) to create a FileOutputStream.

Hide Answer
▼17.4.5
What will happen if you attempt to create an input stream on a nonexistent file? What will happen if you attempt to create an output stream on an existing file? Can you append data to an existing file?
A FileNotFoundException would occur if you attempt to create an input stream for a nonexistent file. You can append data in an existent file if the output stream is created using new FileOutputStream(filename, true) or new FileOutputStream(File, true). Otherwise, the file is overridden if it already exists.

Hide Answer
▼17.4.6
How do you append data to an existing text file using java.io.PrintWriter?
An instance of FileInputStream can be used as an argument to construct a Scanner and an instance of FileOutputStream can be used as an argument to construct a Formatter. So you can create a Formatter to append text into a file using
```
new Formatter(new FileOutputStream("temp.txt", true));
```
If temp.txt does not exist, it is created. If temp.txt already exists, new data is appended to the file.

Hide Answer
▼17.4.7
Suppose a file contains an unspecified number of double values that were written to the file using the writeDouble method using a DataOutputStream, how do you write a program to read all these values? How do you detect the end of a file?
You can use the readDouble method in the DataInputSteam to read double values and use the EOFException to detect the end of the file.

Hide Answer
▼17.4.8
What is written to a file using writeByte(91) on a FileOutputStream?
writeByte(91) writes one byte for number 91 (0x5B in hex, 01011011 in binary) is written to a file using FileOutputStream.

Hide Answer

▼17.4.9

How do you check the end of a file in an input stream (FileInputStream, DataInputStream)?

If the read() method in InputStream returns -1, it is at the end of the file.

Hide Answer

▼17.4.10

What is wrong in the following code?

```
import java.io.*;

public class Test {
  public static void main(String[] args) {
    try (
      FileInputStream fis = new FileInputStream("test.dat"); ) {
    }
    catch (IOException ex) {
      ex.printStackTrace();
    }
    catch (FileNotFoundException ex) {
      ex.printStackTrace();
    }
  }
}
```

Since java.io.FileNotFoundException is a subclass of IOException, the catch clause for java.io.FileNotFoundException should be put before the catch clause for java.io.IOException.

Hide Answer

▼17.4.11

Suppose you run the following program on Windows using the default ASCII encoding after the program is finished, how many bytes are there in the file t.txt? Show the contents of each byte.

```
public class Test {
  public static void main(String[] args)
      throws java.io.IOException {
    try (java.io.PrintWriter output =
        new java.io.PrintWriter("t.txt"); ) {
      output.printf("%s", "1234");
      output.printf("%s", "5678");
      output.close();
    }
  }
}
```

Java uses Unicode, but Windows uses ASCII. The Unicode is converted to ASCII code when writing a character. After the program is finished, the file will contain eight bytes, each represents an ASCII code. So, the values are

```
31 32 33 34 35 36 37 38
```

Note the ASCII code in hex for character 1 is 31.

Hide Answer

▼17.4.12

After the following program is finished, how many bytes are there in the file t.dat? Show the contents of each byte.

```
import java.io.*;

public class Test {
  public static void main(String[] args) throws IOException {
    try (DataOutputStream output = new DataOutputStream(
        new FileOutputStream("t.dat")); ) {
      output.writeInt(1234);
      output.writeInt(5678);
      output.close();
    }
  }
}
```

Each int value takes four bytes. Since two int values are written into the file, the file contains eight bytes. The values are

```
00 00 04 D2 00 00 16 2E
```

The first four bytes are for 1234, which equals to 4D2 in hex, and the second byte is for 5678, which equals to 2246 in hex.

Hide Answer

▼17.4.13

For each of the following statements on a DataOutputStream output, how many bytes are sent to the output?

```
output.writeChar('A');
output.writeChars("BC");
output.writeUTF("DEF");
```

output.writeChar('A'); => 2 bytes
output.writeChars("BC"); => 4 bytes
output.writeUTF("DEF"); => 2 + 3 bytes (the first two bytes store the number of characters in the string. Each ASCII character takes one byte in UTF)

Hide Answer

▼17.4.14

What are the advantages of using buffered streams? Are the following statements correct?

```
BufferedInputStream input1 =
  new BufferedInputStream(new FileInputStream("t.dat"));

DataInputStream input2 = new DataInputStream(
  new BufferedInputStream(new FileInputStream("t.dat")));

DataOutputStream output = new DataOutputStream(
  new BufferedOutputStream(new FileOutnputStream("t.dat")));
```

Since physical input and output involving I/O devices are typically very slow compared with CPU processing speeds, buffered input/output streams can be used to improve performance. You can create a buffered input stream by wrapping a BufferedInputStream/BufferedReader on any instance of InputStream/Reader, and create a buffered output stream by wrapping a BufferedOutputStream/BufferedWriter on any instance of BufferedOutputStream/Writer.

Hide Answer

Section 17.5

▼17.5.1

How does the program check if a file already exists?

Create a File object and use its exists() method to test if a file exists.

Hide Answer

▼17.5.2

How does the program detect the end of the file while reading data?

If the read() method returns -1, it indicates the end of file.

Hide Answer

▼17.5.3

How does the program count the number of bytes read from the file?

Every time a byte is read, numberOfBytesCopied is incremented by 1.

Hide Answer

Section 17.6

▼17.6.1

What types of objects can be stored using the ObjectOutputStream? What is the method for writing an object? What is the method for reading an object? What is the return type of the method that reads an object from ObjectInputStream?

Any objects that are instance of Serializable may be stored using the object stream. You use the writeObject method to write an object to the object output stream and use readObject to read an object from the object input stream. The readObject method returns a value of the Object type.

Hide Answer

▼17.6.2

If you serialize two objects of the same type, will they take the same amount of space? If not, give an example.

No. For example, two ArrayList objects may have different size, so they are serialized into different sizes. Consider two JButton objects. If one button has an icon, and the other does not, the one with an icon will require more space in the file when it is serialized.

Hide Answer

▼17.6.3

Is it true that any instance of java.io.Serializable can be successfully serialized? Are the static variables in an object serialized? How do you mark an instance variable not to be serialized?

An object may not be serialized even though its class implements java.io.Serializable, because it may contain non-serializable instance variables. Implementing java.io.Serializable is a necessary requirement for serialization, but not sufficient. You still have to ensure that all the variables in the object are serializable. A static variable is not serialized. If you don't want a variable to be serialized, mark it transient.

Hide Answer

▼17.6.4

Can you write an array to an ObjectOutputStream?

Yes. An array can be serialized if all its elements can be serialized.

Hide Answer

▼17.6.5

Is it true that DataInputStream/DataOutputStream can always be replaced by ObjectInputStream/ObjectOutputStream?

Yes. Because ObjectInputStream/ObjectOutputStream contains all features and operations in DataInputStream/DataOutputStream.

Hide Answer

▼17.6.6

What will happen when you attempt to run the following code?

```
import java.io.*;
public class Test {
  public static void main(String[] args) throws IOException {
    try ( ObjectOutputStream output =
        new ObjectOutputStream(new FileOutputStream("object.dat")); )
{
      output.writeObject(new A());
    }
  }
}

class A implements Serializable {
  B b = new B();
}

class B {
}
```

A java.io.NotSerializableException would occur.

Hide Answer

Section 17.7

▼17.7.1

Can RandomAccessFile streams read and write a data file created by DataOutputStream? Can RandomAccessFile streams read and write objects?
Yes, because they share the same interface for reading and writing data in the same format. No. Cannot write objects.

Hide Answer

▼17.7.2

Create a RandomAccessFile stream for the file address.dat to allow the updating of student information in the file. Create a DataOutputStream for the file address.dat. Explain the differences between these two statements.

```
RandomAccessFile raf = new RandomAccessFile("address.dat", "rw");
DataOutputStream outfile = new DataOutputStream(
  new FileWriter("address.dat"));
```

To create a RandomAccessFile stream, you simply use the RandomAccessFile constructor. To create a DataOutputStream, you use DataOutputStream wrapped on FileOutputStream.

Hide Answer

▼17.7.3

What happens if the file test.dat does not exist when you attempt to compile and run the following code?

```
import java.io.*;
public class Test {
  public static void main(String[] args) {
    try ( RandomAccessFile raf =
        new RandomAccessFile("test.dat", "r"); ) {
      int i = raf.readInt();
    }
    catch (IOException ex) {
      System.out.println("IO exception");
    }
  }
}
```

It will compile fine, but raises a run time exception on invoking readInt() because nothing is in the file.

Hide Answer