KEY

1. (5 points)Write code that opens a file named NumberList.txt for writing, but does not erase the file’s contents if it already exists.

ANS:

FileWriter fwriter =  
 new FileWriter("NumberList.txt", true);

PrintWriter outFile = new PrintWriter(fwriter);

(Question 2 to 4: 5points each, total 15 points)

1. (5 points)Write code that does the following: opens a file named NumberList.txt, uses a loop to write the numbers 1 through 100 to the file, and then closes the file.

ANS:

PrintWriter outFile =

new PrintWriter("NumberList.txt");  
  
for (int i = 1; i <= 100; i++)  
 outFile.println(i);  
outFile.close();

1. (5 points)Write code that does the following: opens the NumberList.txt file that was created by the code in Question 2, reads all of the numbers from the file and displays tem, and then closes the file.

ANS:

File file = new File("NumberList.txt");  
Scanner inFile = new Scanner(file);  
while (inFile.hasNext())  
{

int number = inFile.nextInt();  
 System.out.println(number);  
}  
inFile.close();

1. (5 points)Modify the code you wrote in Question 3 so it adds all of the numbers read from the file and displays their total.

ANS:

File file = new File("NumberList.txt");  
Scanner inFile = new Scanner(file);  
int total = 0;  
while (inFile.hasNext())  
{  
 int number = inFile.nextInt();  
 total += number;  
}  
inFile.close();

System.out.println("The total is " + total);

1. (4 points) This class allows you to read a line from a file.
2. FileWriter
3. Scanner
4. InputFile
5. FileReader

ANS: b

1. (4 points)This class allows you to use the print and println methods to write data to a file.
2. File
3. FileReader
4. OutputFile
5. PrintWriter

ANS: d

1. (4 points) When a program is finished using a file, it should do this:
2. Erase the file
3. Close the file
4. Throw an exception
5. Reset the read position

ANS: b

1. (4 points)To open a file for reading, you use the following classes.
2. File and Writer
3. File and Output
4. File and Input
5. File and Scanner

ANS: d

1. (4 points)To open a file for writing, you use the following class.
2. PrintWriter
3. FileOpen
4. OutputFile
5. FileReader

ANS: a

1. (4 points) (True/False) When in the same try statement you are handling multiple exceptions an some of the exceptions are related to each other through inheritance, you should handle the more general exception classes before the more specialized exception classes.

ANS: false

1. (4 points) (True/False) When an exception is thrown by code inside a try block, all of the statements in the try block are always executed.

ANS: false

1. (4 points)(True/False) IOException serves as a superclass for exceptions that are related to programming errors, such as an out-of-bounds array subsdcript.

ANS: false

1. (4 points)(True/False) The throws clause causes an exception to be thrown.

ANS: False

1. (2 points) Find the error

catch (fileNotFoundException e)  
{  
 System.out.println("File not found.");  
}  
try  
{  
 File file = new File("MyFile.txt");  
 Scanner inputFile = new Scanner(file);  
}

ANS: The try block must appear first.

1. (2 points) Find the error

//Assume inputFile references a Scanner object  
try  
{  
 input = inputFile.nextInt();  
}  
finally  
{  
 inputFile.close();  
}  
catch (inputMismatchException e)  
{  
 System.out.println(e.getMessage());  
}

ANS: The finally block must appear after the catch blocks.

1. (4 points) When an exception is generated, it is said to have been \_\_\_\_\_\_\_\_\_\_.
2. built
3. thrown
4. caught
5. killed

ANS: b

1. (4 points)This is a section of code that gracefully responds to exceptions.
2. exception generator
3. exception manipulator
4. exception handler
5. exception monitor

ANS: c

1. (4 points)If your code does not handle an exception when it is thrown, it is dealt with by this.
2. default exception handler
3. the operating system
4. system debugger
5. default exception generator

Ans: a

1. (4 points)All exception classes inherit from this class.
2. Error
3. RuntimeException
4. JavaException
5. Throwable

ANS: d

1. (4 points) FileNotFoundException inherits from \_\_\_\_\_\_\_\_\_.
2. Error
3. IOException
4. JavaException
5. FileException

ANS: b

21-22 (5 points each)

1. (5 points)Write an exception class that can be thrown when a negative number is passed to a method.

ANS:

public class NegativeNumber extends Exception

{

/\*\*

No-arg constructor

\*/

public NegativeNumber()

{

super("Error: Negative number");

}

/\*\*

The following constructor accepts the number that

caused the exception.

@param n The number.

\*/

public NegativeNumber(int n)

{

super("Error: Negative number: " + n);

}

}

1. (5 points)Write a statement that throws an instance of the exception class that you created in above problem.

ANS:

throw new NegativeNumber();

Or

throw new NegativeNumber(number);

23-24 (5 points each)

1. (5 points)The method getValueFromFile is public and returns an int. It accepts no arguments. The method is capable of throwing an IOException and a FileNotFoundException. Write the header for this method.

ANS:

public int getValueFromFile() throws IOException,

FileNotFoundException

1. (5 points)Write a try statement that calls the getValueFromFile method described in above problem. Be sure to handle all the exceptions that the method can throw.

ANS:

try

{

value = getValueFromFile();

}

catch (FileNotFoundException e)

{

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

}

catch (IOException e)

{

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

}