COMPSCI 230 Tutorial 3 - Answers

Java: Exception handling & FileIO

Today we are going to continue exception handling from tutorial 2 and will also learn about file operations like read. This tutorial will help you understand how to perform file reading with exception handling.

1. Remind ourselves of what the following keywords mean:

try	**Creates a guarded region, that can cause an exception in program.**
catch	**Handles the exception that is thrown by try block. There can be multiple catch blocks.**
throw	**It is used to explicitly throw an exception.**
throws	**It is a list of exceptions that a function can throw, this list helps caller function to be aware of which exceptions to handle.**
finally	**This block always gets executed whether or not exception occurs.**

2. (a) Tut3_01 class should read a file and compute the sum & average of the numbers found in the file. Using exception handling display appropriate error messages:

- If file does not exist

- If input file does not contain a numeric value

Note: Download numbers01.txt from canvas

Excepted Output is as below.

Success scenario:

```
Sum  is: 220.95
Avg. is: 24.55
```

Erroneous scenario:

```
Invalid input

File couldn't be open
```

```java
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.InputMismatchException;
import java.util.Scanner;

public class Tut3_01 {

	public static void main(String[] args) throws IOException {
		FileReader fr = null;
		Scanner in = null;
		double sum = 0d;
		int count = 0;
		try {
			fr = new FileReader("numbers02.txt");
			in = new Scanner(fr);
			while (in.hasNext()) {
				sum += in.nextDouble();
				count++;
			}
			System.out.println("Sum  is: " + sum);
			System.out.format("Avg. is: %.2f%n", sum/count);
			// %.2f display floating point number with 2 digits after
		decimal point
			// %n is for new line
			// or System.out.println("Avg. is: " + Math.round(sum /
count * 100) / 100d);
		} catch (FileNotFoundException e) {
			System.out.print("File couldn't be open");
		} catch (InputMismatchException e) {
			System.out.print("Invalid input");
		} finally {
			if (in != null) {
				in.close();
			}
			if (fr != null) {
				fr.close();
			}

		}
	}

}
```

2. (b) Modify above class to handle non numeric input and continue to read file if numeric data is still present in the file. At the end, display the count of non-numeric data in file with sum and average of numeric data as in above question.

Hint: Use nested try-catch statements.

Note: Download numbers02.txt from canvas

Excepted Output:

```
Sum  is: 267.00
Avg. is: 17.80
Non numeric: 4
```

```java
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.InputMismatchException;
import java.util.Scanner;

public class Tut3_02 {

	public static void main(String[] args) throws IOException {
		FileReader fr = null;
		Scanner in = null;
		double sum = 0d;
		int count = 0;
		int non_num_count = 0;
		try {
			fr = new FileReader("numbers02.txt");
			in = new Scanner(fr);
			while (in.hasNext()) {
				try {
					sum += in.nextDouble();
					count++;
				} catch (InputMismatchException e) {
					in.next();
					non_num_count++;
				}
			}
			System.out.format("Sum  is: %.2f%n", sum);
			System.out.format("Avg. is: %.2f%n", sum / count);
			System.out.println("Non numeric: " + non_num_count);
		} catch (FileNotFoundException e) {
			System.out.print("File couldn't be open");
		} finally {
			if (in != null) {
				in.close();
			}
			if (fr != null) {
				fr.close();
			}

		}
	}

}
```

3. Quiz

3.1. What method of an Exception object prints a list of methods that were called before the exception was thrown?

a. printErrors()

b. getMessage()

c. printStackTrace()

d. traceStack()

Answer: C

3.2. What method of an Exception object returns a message string?

a. getError()

b. getMessage()

c. printMessage()

d. traceMessage()

Answer: B

3.3. What happens during execution if a negative value is used for an array index?

a. An IndexOutOfBoundsException is thrown.

b. A NumberFormatException is thrown.

c. The first slot of the array is used.

d. This is an Error, so the program immediately terminates no matter what.

Answer: A

3.4. A method must do either of two things with a checked exception. What are those two things?

a. (1) Ignore the exception, or (2) check the exception.

b. (1) Handle the exception in a catch{} block, or (2) return the exception to the sender.

c. (1) Handle the exception in a catch{} block, or (2) throw the exception to the method that called this method.

d. (1) Handle the exception in a catch{} block, or (2) handle the exception in a try{} block,

Answer: C

3.5. What is the only type of exception that is NOT checked?

a. Class Exception.

b. Class RunTimeException and its subclasses.

c. Class IOException and its subclasses.

d. Class ArithmeticException only.

Answer: B

3.6. Say that methodA calls methodB, and methodB calls methodC. MethodC might throw a NumberFormatException. Can the program be written so that methodA handles the exception?

a. No, the exception must be handled by methodC or its caller, methodB.

b. No, methodC must handle the exception.

c. Yes, if the header for methodC says ...throws NumberFormatException

d. Yes, if the headers for methodC and methodB say ...throws NumberFormatException

Answer: D

3.7. Which of the following is a correct outline of a method that throws an exception when it finds a problem?

a.

```
public void someMethod ()
{
 ...
 if ( problem )
   throw new Exception("Useful Message");
 ...
}
```

b.

```
public void someMethod () throws Exception
{
 ...
 if ( problem )
   Exception("Useful Message");
 ...
}
```

c.

```
public void someMethod () throws Exception
{
  ...
  if ( problem )
    throw( new Exception("Useful Message") );
  ...
}
```

d.

```
public void someMethod () throws Exception
{
  ...
  if ( problem )
    throw new Exception("Useful Message");
  ...
}
```

Answer: D

3.8. MethodX might encounter an IOException or an AWTException, but handles neither. How should the header for methodX be written?

a. ... methodX(...) throws IOException, AWTException

b. ... methodX(...) throws IOException or AWTException

c. ... methodX(...) throws IOException throws AWTException

d. ... methodX(...) throws (IOException, AWTException)

Answer: A

3.9. You are writing a program to check people into a hotel. People over 65 get a 10% discount. How would you write the program?

a. Use an if statement to test for age 65 or greater. If one is detected, throw an exception. A catch{} block will apply the discount.

b. Normal if-else programming logic will be used.

c. Subtract 65 from the person's age and divide the bill by the result. If an ArithmeticException exception is thrown, apply the 10% discount in the exception handler.

d. Create an array with 120 elements, one element for each possible hotel guest age. Each element contains the discount rate for that age.

Answer: B

3.10. Say that a method catches an IOException in a catch{} block. Is it possible for that block to do some processing and then throw the same exception to the caller?

a. No---when an exception is caught the Exception object is destroyed.

b. No---an exception may be caught only once.

c. Yes---as long as the method also has a throws clause for that exception.

d. Yes---but the catch{} block must construct a new Exception object.

Answer: C