

COE 312 – Data Structures

Welcome to Exam I
Monday October 26, 2015

Instructor: Dr. Wissam F. Fawaz

Name: _____

Student ID: _____

Instructions:

1. This exam is **Closed Book**. Please do not forget to write your name and ID on the first page.
2. You have exactly **55 minutes** to complete the **4** required problems.
3. Read each problem carefully. If something appears ambiguous, please write your assumptions.
4. Points allocated to each problem are shown in square brackets.
5. Put your answers in the space provided only. No other spaces will be graded or even looked at.

Good Luck!!

Problem 1: Exception Handling and Recursion (10 minutes) [20 points]

- 1) Consider the following recursive method definition:

```
public String foo(String str) {
    if(str.length() == 0)
        return "";
    else {
        String str_mod = str.substring(0, str.length()-1);
        return str.charAt(str.length()-1)+foo(str_mod);
    }
}
```

What output does `foo` produce if a value of "funny" is passed as the parameter?

- fun
 - ynuf
 - funny
 - None of the above**
- 2) What is wrong with the following recursive `sum` method? The method is supposed to sum up the values between 1 and `x` (for instance, `sum(5)` should return $5+4+3+2+1=15$).

```
public int sum(int x) {
    if(x==0) return 0;
    else return x+sum(x-1);
}
```

- The base case should return 1 instead of 0
 - The base case condition should be $(x \leq 0)$ instead of $(x == 0)$**
 - The recursive part should return `sum(x) + 1`
 - The method should return a boolean instead of an int
- 3) If the code in a method can potentially throw a checked exception, then that method must
- handle the exception
 - have a throws clause listed in the method header
 - Neither (a) nor (b)
 - Either (a) or (b)**
- 4) Which of the following methods are part of the `Exception` class and can be used to give more information about a thrown exception?
- `printInfoTrace`
 - `getMessage`
 - `getStackTrace`
 - Both (b) and (c)**

- 5) How many times will the following method print the statement "Print this line." out, if a value of 10 is passed as the parameter?

```
public void message(int n) {
    if(n > 0) {
        System.out.println("Print this line.");
        message(n - 1);}
}
```

- a. 9
 - b. 10
 - c. 8
 - d. An infinite number of time
- 6) Which of the following is an unchecked exception?
- a. RuntimeException
 - b. ClassCastException
 - c. **Both of the above**
 - d. None of the above
- 7) The number of method calls to recursively calculate the Fibonacci value of 5 is
- a. 16
 - b. 14
 - c. 12
 - d. **None of the above**
- 8) Which of the following statements throws a `NumberFormatException` if the `String str` variable contains a sequence of alphabetic characters?
- a. `int value = Integer.parseInt(sr);`
 - b. `double value = Double.parseDouble(str);`
 - c. **Both of the above**
 - d. None of the above
- 9) Which of the following statements generates an `ArithmeticException`?
- a. `System.out.println(10/0.0);`
 - b. `System.out.println(0/0.0);`
 - c. Both of the above
 - d. **None of the above**
- 10) Consider the following method declaration:
- ```
void exam(int i) {
 if(i>1) {
 exam(i/2);
 exam(i/2);
 }
 System.out.print("*");
}
```
- How many asterisks are printed by the method call `exam(5)`?
- a. 3
  - b. 4
  - c. **7**
  - d. None of the above

**Problem 2:** Polymorphism and NIO/IO classes (10 minutes) [20 points]

- 1) Which of the following is not an input byte-based stream?
  - a. `FileOutputStream`
  - b. `FileReader`
  - c. **Both of the above**
  - d. None of the above
  
- 2) Which of the following classes provides static methods for common file and directory manipulations, including methods for copying files, creating and deleting files and directories, getting information about files and directories, and reading the contents of files and directories?
  - a. `File`
  - b. `DirectoryStream`
  - c. `Paths`
  - d. **None of the above**
  
- 3) Which of the following interfaces of the `java.io` package can be used to filter files based on their extensions?
  - a. **`FileFilter`**
  - b. `Filter`
  - c. All of the above
  - d. None of the above
  
- 4) Which of the following statements can be used to create an object that can perform character-based input from a webpage represented by a `URL` object called `url`?
  - a. `BufferedReader br = new BufferedReader(url.openStream());`
  - b. `BufferedReader br = new InputStreamReader(url.openStream());`
  - c. Both of the above
  - d. **None of the above**
  
- 5) The call to an overloaded method is bound to a method definition during
  - a. **Compilation**
  - b. Execution
  - c. Debugging
  - d. None of the above
  
- 6) Let `Speaker` be an interface that has a single method called `speak()`. Suppose that `Angry` and `Politician` are two classes that implement the `Speaker` interface. Assume also that `Angry` has a method called `shout()` along with a 0-parameter constructor and that `Politician` has a method called `pontificate()` along with a 0-parameter constructor. Now, consider the following code:
 

```
Speaker speaker = new Angry();
((Politician) speaker).pontificate();
```

 Which of the following is true?
  - a. This code will result in a compile-time error
  - b. **This code will result in a run-time error**
  - c. The `pontificate` method as defined in the `Politician` class will be called
  - d. None of the above is true

- 7) Consider the `Speaker` interface along with both the `Angry` and `Politician` classes described in the previous question. Suppose the following code fragment is executed:

```
Speaker speaker = new Angry();
((Angry) speaker).shout();
speaker = new Politician();
speaker.pontificate();
```

Which of the following is true?

- a. **This code will result in a compile-time error**
  - b. This code will result in a run-time error
  - c. The `shout` method as defined in the `Angry` class will be called and then the `pontificate` method as defined in the `Politician` class will be called
  - d. None of the above is true
- 8) Which statement best describes the relationship between superclass and subclass types?
- a. A subclass object cannot be assigned to a superclass variable and a superclass object cannot be assigned to a subclass variable.
  - b. A subclass object can be assigned to a superclass variable and a superclass object can be assigned to a subclass variable.
  - c. A superclass object can be assigned to a subclass variable, but a subclass object cannot be assigned to a superclass variable.
  - d. **A subclass object can be assigned to a superclass variable, but a superclass object cannot be assigned to a subclass variable.**

- 9) Suppose that the class `Rodent` has a child class called `Rat` and another child class called `Mouse`. Assume also that the class `Mouse` has a child class called `PocketMouse`. Examine the following:

```
Rodent rod;
Rat rat = new Rat();
Mouse mouse = new Mouse();
PocketMouse pkt = new PocketMouse();
```

Which of the following will cause a runtime error?

- a. `rod = rat;`
  - b. `rod = pkt;`
  - c. `pkt = rat;`
  - d. **None of the above**
- 10) Say that the situation is the same as in the previous question. Which of the following statements correctly creates an array that can hold both `Mouse` and `PocketMouse` objects?
- a. `Mouse[] arr = new Mouse[10];`
  - b. `Rodent[] arr = new Rodent[10];`
  - c. **All of the above**
  - d. None of the above

**Problem 3: I/O streams (15 minutes) [30 points]**

Design and implement a Java program that copies the content of a binary file called “srcFile.jpg” into a target file called “trgtFile1.jpg” using the classes of the old `java.io` package. Your program should then copy the content of “srcFile.jpg” into another target file called “trgtFile2.jpg” using the classes of the new `java.nio` package. Assume in both cases that the source and the target files belong to the same directory as your Java project.

```
import java.io.*;
import java.nio.file.*;
public class BinaryFileCopy {
public static void main(String[] args) throws Exception{
 File srcFile = new File("srcFile.jpg");
 File trgtFile = new File("trgtFile1.jpg");
 Path srcPath = Paths.get("srcFile.jpg");
 Path trgtPath = Paths.get("trgtFile2.jpg");
 FileInputStream fis = new FileInputStream(srcFile);
 FileOutputStream fos = new FileOutputStream(trgtFile);
 byte[] buf = new byte[1024];
 int len;
 while((len=fis.read(buf))>0)
 fos.write(buf, 0, len);
 fis.close(); fos.close();
 Files.copy(srcPath, trgtPath);
}
}
```

### Problem 4: JDOM (20 minutes) [30 points]

Consider the following XML and JSON documents that show credit client details:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>&lt;clients&gt;   &lt;client&gt;     &lt;client-name&gt;Steve&lt;/client-name&gt;     &lt;account-number&gt;100&lt;/account- number&gt;     &lt;client-balance&gt;-345.67&lt;/client-balance&gt;   &lt;/client&gt;   &lt;client&gt;     &lt;client-name&gt;Sam&lt;/client-name&gt;     &lt;account-number&gt;200&lt;/account- number&gt;     &lt;client-balance&gt;-42.16&lt;/client-balance&gt;   &lt;/client&gt;   &lt;client&gt;     &lt;client-name&gt;Bob&lt;/client-name&gt;     &lt;account-number&gt;300&lt;/account- number&gt;     &lt;client-balance&gt;224.62&lt;/client-balance&gt;   &lt;/client&gt; &lt;/clients&gt;</pre> | <pre>[   {“client-name”:“Steve”,    “account-number”:100,    “client-balance”:345.67   },   {“client-name”:“Sam”,    “account-number”:200,    “client-balance”:42.16   },   {“client-name”:“Bob”,    “account-number”:300,    “client-balance”:224.62   } ]</pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

The information given on the left hand side is stored in an **XML file** called “clients.xml” while the information on the right hand side is stored in a **JSON file** called “clients.json”. Both files reside inside **folders** named “XML” and “JSON” respectively on the “<http://www.wissamfawaz.com>” **webserver**. So, “clients.xml” can be referenced through the following URL: “<http://www.wissamfawaz.com/XML/clients.xml>” while “clients.json” can be referenced via: “<http://www.wissamfawaz.com/JSON/clients.json>”. These xml and JSON files might be used in an accounts receivable system to keep track of the amounts owed to a company by its credit clients.

- Using the JSON parser, write a Java program that outputs the total amount of money the clients owe the company by summing up the “credit-balance” values pertaining to all the clients.

```
import java.io.InputStream;
import java.net.URI;
import java.net.URL;
import org.json.JSONArray;
import org.json.JSONObject;
import org.json.JSONTokener;
public class ClientDetailsParsing {
 public static void main(String[] args) {
 URL url = //Incomplete
 InputStream is = url.openStream();
 JSONTokener tokenizer = new JSONTokener(is);
 JSONArray clientsArray = new JSONArray(tokenizer);
 JSONObject clientObject;
 double total=0.0, balance;
 }
}
```

```

 for(int i=0; i<clientsArray.length(); i++) {
 clientObject = clientsArray.getJSONObject(i);
 balance=clientObject.getDouble("client-balance");
 total+=balance;
 }
 System.out.println("Total amount:" + total);
 } // end main method
} // end ClientDetails class

```

2. Using the **JDOM parser**, write a Java program that:
  - a. outputs the number of clients having a negative credit balance, and then
  - b. outputs the "client-name" of the client having the largest "credit-balance" associated with him.

```

import java.io.InputStream;
import java.net.URL;
import java.util.List;
import org.jdom2.Document;
import org.jdom2.Element;
import org.jdom2.input.SAXBuilder;
import org.jdom2.filter.ElementFilter;
import org.jdom2.util.IteratorIterable;
public class ParsingCreditClientInformation {
 public static void main(String[] args)throws Exception {
 URL url = //Incomplete
 InputStream is = url.openStream();
 SAXBuilder builder = new SAXBuilder();
 Document doc = builder.build(is);
 Element root = doc.getRootElement();
 IteratorIterable<Element> balanceIterator =
 root.getDescendants(
 new ElementFilter("client-balance"));
 Element clientBalance; double balance;
 int nbNegatives=0;
 double largestBalance=Double.MIN_VALUE;
 while(balanceIterator.hasNext()) {
 clientBalance=balanceIterator.next();
 balance=Double.parseDouble(
 clientBalance.getValue());
 if(balance<0) nbNegatives++;
 if(balance>largestBalance)
 largestBalance = balance;
 }
 System.out.println("Nb of negatives:"+nbOfNegatives);
 System.out.println("Largest balance:"+largestBalance);
 } // end main method
} // end ParsingCreditClientInformation class

```



## Appendix: Classes and Methods

### 1. JDOM parser related classes along with their associated methods:

|                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><u>org.jdom2.Document</u></p> <ul style="list-style-type: none"> <li>• Element getRootElement()</li> </ul> <p><u>org.jdom2.Element</u></p> <ul style="list-style-type: none"> <li>• List&lt;Element&gt; getChildren(String)</li> <li>• IteratorIterable&lt;Element&gt; getDescendants(ElementFilter)</li> <li>• String getText()</li> <li>• String getChildText(String)</li> </ul> | <p><u>org.jdom2.filter.ElementFilter</u></p> <ul style="list-style-type: none"> <li>• ElementFilter(String)</li> </ul> <p><u>org.jdom2.util.IteratorIterable</u></p> <ul style="list-style-type: none"> <li>• boolean hasNext()</li> <li>• Element next()</li> </ul> <p><u>java.util.List&lt;Element&gt;</u></p> <ul style="list-style-type: none"> <li>• int size()</li> <li>• Element get(int index)</li> </ul> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 2. Classes of the java.io package:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><u>java.io.File</u></p> <ul style="list-style-type: none"> <li>• File(String)</li> <li>• long length()</li> </ul> <p><u>java.io.FileOutputStream</u></p> <ul style="list-style-type: none"> <li>• FileOutputStream(File)</li> <li>• void write(byte[] buff, int off, int len)</li> <li>• void close()</li> </ul> <p><u>java.io.FileInputStream</u></p> <ul style="list-style-type: none"> <li>• FileInputStream(File)</li> <li>• int read(byte[])</li> <li>• void close()</li> </ul> | <p><u>java.io.FileWriter</u></p> <ul style="list-style-type: none"> <li>• FileWriter(File)</li> </ul> <p><u>java.io.PrintWriter</u></p> <ul style="list-style-type: none"> <li>• PrintWriter(BufferedWriter)</li> <li>• void print(String)</li> <li>• void println(String)</li> <li>• void close()</li> </ul> <p><u>java.io.InputStreamReader</u></p> <ul style="list-style-type: none"> <li>• InputStreamReader(InputStream)</li> </ul> <p><u>java.io.BufferedReader</u></p> <ul style="list-style-type: none"> <li>• BufferedReader(InputStreamReader)</li> <li>• String readLine()</li> </ul> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 3. Classes of the java.nio package:

|                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><u>java.nio.file.Files</u></p> <ul style="list-style-type: none"> <li>• static BufferedReader newBufferedReader(Path)</li> <li>• static BufferedWriter newBufferedWriter(Path)</li> <li>• static InputStream newInputStream(Path)</li> <li>• static Path copy(Path src, Path trgt)</li> <li>• static void write(Path trgt, byte[] buff)</li> </ul> | <p><u>java.nio.file.Paths</u></p> <ul style="list-style-type: none"> <li>• static Path get(String)</li> </ul> <p><u>java.nio.file.Path</u></p> <ul style="list-style-type: none"> <li>• Path getFileName()</li> <li>• File toFile()</li> </ul> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 4. JSON parser classes and some of their related methods.

|                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><u>JSONArray</u></p> <p>JSONArray(JSNTokener)</p> <p>int length()</p> <p>JSONObject getJSONObject(int i)</p> <p>JSONArray getJSONArray(int i)</p> <p>double getDouble(int i)</p> <p>int getInt(int i)</p> <p>String getString(int i)</p> | <p><u>JSONObject</u></p> <p>JSONObject(JSNTokener)</p> <p>JSONArray getJSONArray(String key)</p> <p>JSONObject getJSONObject(String key)</p> <p>String getString(String key)</p> <p>double getDouble(String key)</p> <p>int getInt(String key)</p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|