1. A two paragraph commentary on the work that each team member completed
   Each team member was able to complete this laboratory assignment at the correct time. Troy was responsible for completing the required deliverables 5 and 6. This deliverables includes providing the output from at least five runs of your benchmark's, demonstrating features and correctness, and a providing a detailed report addressing the performance trade-offs associated with serialization. Troy was able to complete this deliverables at the correct time in order to hand in this assignment on time. He was also responsible working with me to be able to work on the final version of the course code for your serialization benchmarking framework.

   I was able to complete this laboratory assignment as well at the correct time. I was responsible for providing a description of the eXtensible markup language and its features, strengths, and weaknesses, and providing an explanation of the default serialization method provided by the Java language. I was working with my group member Troy to work on the source code to test whether the xstream or the Java serialization is quicker to serialize an object.

2. A description of the eXtensible markup language and its features, strengths, and weaknesses.

   Features

   - XML files are text files which means it can be modified by any text editor.
   - provides a way for communication between applications.
   - A way to describe data structures, or files.

   Strengths

   - It is platform-independent so it is relatively immune to changes in technology.
   - it manifests as a plain text file.
   - XML has the ability to represent the most general data structures such as records, lists, and trees.
   - It is heavily used as a format for document storage and processing, both online and offline.

   Weaknesses

   - The syntax of XML is considered to to expressed in more words than are needed.
   - The basic parsing requirements do not support a very wide array of data types.
   - The Syntax contains unnecessary features
   - Modeling overlapping data structures requires extra effort.
   - There are other ways to serialize structured data that are more popular to use that are smaller and faster. JSON and Protocol Buffers are two of them.

3. An explanation of the default serialization method provided by the Java language. The default serialization method provided by the Java language is able to serialize objects. The process of serialization is a conversion of an object into a series of bytes. This conversion process allows an object can be useful for being able to transmit data of an object across the network. One example of these networks could be a JVM (Java Virtual Machine).

```
     String parentheticTree =
          parentheticRepresentation(randomTree,randomTree.getRoot());
try{
         FileOutputStream fileOut =
         new FileOutputStream("serializer.txt");
         ObjectOutputStream out = new
             ObjectOutputStream(fileOut);
```

```
           out.writeObject(parentheticTree);
           out.close();
           fileOut.close();
           System.out.printf("Serialized data is saved in
               serializer.txt");
        }catch(IOException i)
        {
            i.printStackTrace();
        }
```

This is the source code we had used to be able to serialize our object. In order to do so, we decided to implement using a try catch block. The FileOutputStream() method will be able to output a file to be able to store the byte representation of the serialized object. In the case, we decided to output into a text file called serizlier.txt as shown in the source code above. After the file has been declared to output to, it is now time to populate the file with data. In order to do so, one would declared the ObjectOutputStream() method. In order to populate the file, the file needs to be called in as a parameter. After the file has been declared, the next step is to write to the object. In order to do so, one would declare the method writeObject() and pass the object that you want to be serialized. In the case, I decided to declared the string paranthenticTree to be called. After I have populated the data into the file, then I would have to close the OutputStream. After I close the OutputStream(), then I would have successfully serialize an object using the default serialization method by the java programming language.

4. The final version of the source code for your serialization benchmarking framework.
5. Output from at least five runs of your benchmarks, demonstrating features and correctness.
6. A detailed report addressing the performance trade-odds associated with serialization.

7. A reflection of the challenges that you faced when completing this laboratory assignment.
   There were a number of challenge that I faced in order to complete this laboratory assignment. Some of these include using the test object to test the performance of the xstream and the default java serialization. One of the challenges I had faced when completing this laboratory assignment was being able to call an object using the java serialization method to change the serialized object into a binary stream. In order to overcome this challenge, I decided to declare the test object into a string and call the string into the default method which I was able to complete and have it working correctly. Another challenge I faced when completing this assignment was being able to understand what was being asked to complete this assignment. I didn't understand whether I needed to create my own tree class or if I needed to do something else. In order to overcome this issue, I decided to ask Professor Kapfhammer what was being asked. After I asked Professor Kapfhammer what was being asked, I was able to completing this assignment effectively.