

Lab 9: Exceptions, Input/Output and Parsing

0. This lab will be due on **WEDNESDAY, July 6th 2005 promptly at 4 p.m.** . This means your lab should be checked off by 4pm on that due date. You are advised to get your lab checked of as soon as you can, and not wait till a few minutes before it is due.
1. This lab will test your knowledge on handling Exceptions, getting input from the user (this was done in lab 4), reading from files, creating files, and writing to files. There will be two independent parts to the lab. In the first part, you will write your own checked exception and modify a provided Payroll class to throw the exception. You shall also demonstrate test code that causes the exception to be thrown and handles the exception. In the second part of the lab, you will read from a given file, and create and write to a new file.

Part 1 – Exceptions

```
/**
 * The Employee Class defines an employee with a
 * name and an age.
 */
public class Employee {

    // Name of employee
    private String name;
    // Age of employee
    private int age;

    /*
     * Employee Constructor
     * @param name name of employee
     * @param age age of employee
     */
    public Employee(String name, int age){
        // Initialize name and age of employee
        this.name = name;
        this.age = age;
    }

}

package Lab9;

import java.util.ArrayList;

/**
 * The Payroll Class keeps track of all the employees that
 * are on the payroll and provides an addEmployee method
 * for adding employees to the payroll.
 */
public class Payroll {
```

```

// ArrayList to keep track of employees
private ArrayList employees;

/*
 * Payroll Constructor
 * @param none
 */
public Payroll(){
    // Initialize employee ArrayList
    employees = new ArrayList();
}

/*
 * Adds employee to the payroll.
 * @param emp employee to add
 */
public void addEmployee(Employee emp){
    // Add employee to the payroll
    employees.add(emp);
}
}

```

Write and throw an `employeeAlreadyAdded` exception in the `Payroll` class. This exception should occur in case the employee you are trying to add is already in the payroll. Note that this code will still work without the exception. Thus you will need to show us the functioning of your exception code during checkoff.

Part 2 – I/O

In the second part of this lab, we will be writing two methods in a class called `FileUtility`:

- (a) The `displayFile` method takes a file path as argument, of type `String`. This method tries to open the given file path and return if an exception occurs. If no exception occurs, the method goes ahead to read the line from the file path and print each piece of information on the line (token) on a new line on the screen. When creating the `StringTokenizer` for this part, use a white space as the delimiters.
- (b) The `writeToFile` method takes as argument a file you want to create, of type `String`. This method creates the new file and then writes your Name and the String “Java is tough but I am loving it” into the file.

The general structure should be:

```

public class FileUtility{
    public static void displayFile(String filename) {
        // Tries to open the given filename.
    }
}

```

```

        // Returns if an exception occurs.
        // Read the line and print each token of info on a new line on the screen
    }

    public static void writeToFile(String filetocreate) {
        // Create the file, and return if an exception occurs.
        // Write you name and the above String in the file.
    }
}

```

2. Complete the body of the displayFile method. Test it using the file lab9.txt** available on chania. The output should be:

```

Today
07/04/05
Strathmore
University

```

3. Complete the body of the writeToFile method. Test it using the displayFile method. The output should be:

```

Firstname
Lastname
Java
is
tough
but
I
am
loving
it

```

4. Be prepared to answer the following questions during checkoff:
 - What exceptions did you handle in the displayFile and makeFile methods and how did you handle them?
 - Which of the above exceptions were checked exceptions (How do you know?)?
 - What was the hardest part of this lab?

** You will have to copy and paste the lab9.txt file into your Documents and Settings in order to simplify your file path.