

## CS 159 Review Exercises

Names: \_\_\_\_\_

1. Create a new exception class named `InvalidStringException`. This should be a checked exception.
2. Complete the following method so that it conforms to the Javadocs.

```
/**
 * Count the total number of characters in all Strings in the provided array
 * of Strings. All strings must have length of at least one.
 *
 * @param strings - The array of strings.
 * @return The character count.
 * @throws InvalidStringException - Thrown if any string has length 0.
 * @throws NullPointerException - Thrown if there are any null strings in
 *         the array.
 */
public static int countCharacters(String[] strings)
```

3. Finish the following constructor for the attached `Car` class according to the Javadoc.

```
/**
 * Create a car from a comma separated sequence of values. The first value
 * must be an integer, and the second value must be a string. For example:
 *
 * "22,Red"
 *
 * @param carString A comma separated string containing car information.
 * @throws IllegalArgumentException If the string does not match the
 *         expected format.
 */
public Car(String carString) throws IllegalArgumentException
{
```

4. Finish the following utility method according to the provided Javadoc:

```
/**
 * Load a set of transportation objects from a text file. The first line of
 * the file will be an integer that describes the number of objects in the
 * file. Each object will be represented using two lines. The first line
 * will be either "CAR" or "HORSE" indicating the object type, and the next
 * line will be string representation of the desired object.
 *
 * If any errors are encountered while processing the file, then a null
 * pointer should be returned.
 *
 * Here is an example:
 *
 * 3
 * CAR
 * 20,Red
 * HORSE
 * Black Beauty
 * CAR
 * 100,Black
 *
 * @param fileName The name of the file to read.
 * @return An array of transportation objects, or null if the file contains
 *         any errors or cannot be read.
 */
public static Object[] loadTransportFile(String fileName)
{
```

5. Finish the following utility method according to the provided Javadoc:

```
/**
 * Print a string representation of each of the objects in the transports
 * array, one per line.
 *
 * @param transports - array of transportation objects.
 */
public static void printTransportArray(Object[] transports)
{
```

```
public class Horse
{
    private String name;

    public Horse(String name)
    {
        this.name = name;
    }

    public String getName()
    {
        return name;
    }

    public String toString()
    {
        return "HORSE- Name: " + name;
    }
}
```

```
public class Car
{
    private int maxSpeed;
    private String color;

    public Car(int maxSpeed, String color)
    {
        this.maxSpeed = maxSpeed;
        this.color = color;
    }

    public int getMaxSpeed()
    {
        return maxSpeed;
    }

    public String getColor()
    {
        return color;
    }

    public String toString()
    {
        return "CAR- Speed: " + maxSpeed + " Color: " + color;
    }
}
```