Day 09 – Exceptional Situations

Chapter 01, Sec 1.3 – Exceptional Situations, Review Problem(s): ***22***

As you work through the below, put comments in your code describing what you're doing and the connections.

First, open a new Java Project, calling it something like CarException.

Then, add a Java Class, called Car.java, and use the code from Moodle (best to do it this way, rather than the Car class you created, so we all have the same code base).

The Car class does not have any mechanism for handling Exceptions. And in Java, we don't just change the existing class (someone else may be using it).

So we create another class, which will be a subclass, CarSafe, which extends Car.

In CarSafe, we can just use super() in the constructors for a generic object and one with input parameters.

Create a TDCarSafe class to test your new class. You can use the previous TDCar as a model, just be sure to update your class references from Car to CarSafe.

Now, we will add Exception handling to our new CarSafe class.

In this example, we will simply throw an exception if a new object is instantiated with an initial mileage which is negative, ie. less than zero.

The first step is to create our own exception class, CarMileageNegativeException, and we will use the textbook and slides as an example.

```
public class CarMileageNegativeException extends Exception
{
       public CarMileageNegativeException()
       {
              super();
       }
       public CarMileageNegativeException(String message)
       {
              super(message);
       }

}
```

Add this class to your Java Project.

We will update our CarSafe constructor to throw the exception when a negative mileage is input.

Note that we can no longer use the super() call for the constructor which takes input.

```
public class CarSafe extends Car
{
        public CarSafe ( double newMiles, String newColor ) throws
CarMileageNegativeException
        {
                if (newMiles<0)
                        throw new CarMileageNegativeException("input mileage:
"+newMiles+" is less than 0");
                else
                {
                        this.miles = newMiles;
                    this.color = newColor;
                }
        }
        /*
        public CarSafe ( double newMiles, String newColor )
        {
                super(newMiles,newColor);
        }
        */
        public CarSafe ( )
        {
                super();
        }

}
```

And finally, we need to update TDCarSafe so that it also throws the exception, since it is calling the CarSafe constructor.

```
// needed for the Scanner class
import java.util.Scanner;

public class TDCarSafe
{
  public static void main (String[] args) throws CarMileageNegativeException
  {
    CarSafe toyota = new CarSafe(23000, "silver");
    System.out.println( toyota.getMiles() + " " + toyota.getColor() );

    CarSafe landRover = new CarSafe(-5000, "white");

    CarSafe ford = new CarSafe();
    System.out.println( ford );  // this uses the toString method
    ford.changeMiles(1300);
    System.out.println( ford );  // this uses the toString method
  }

}
```