

LCS-ES Group Meeting #7

Exercise 1: The Car Class

This exercise will help you to understand what happens when objects are declared and created, and when methods are called (messages are sent).

It will also help you to understand the difference between copying from one variable to another when the variable is an object and when it is a primitive type (int, double, boolean, etc).

First, take a look at the Car class defined on the next page.

Now “execute” the following code fragment; let one person play the role of each variable (myCar, yourCar, oldSpeed, and newSpeed), have one person draw the memory diagrams (boxes), and let another person be in charge of writing the output values on the board.

```
Car myCar, yourCar;
int oldSpeed, currSpeed;

myCar = new Car("beep");
currSpeed = myCar.getCurrSpeed()

while (currSpeed < 10)
{
    oldSpeed = currSpeed;
    myCar.accelerate(7);
    currSpeed = myCar.getCurrSpeed();
    if ((currSpeed % 2) == 0)
    {
        myCar.blowHorn(currSpeed/5);
    }
    else
    {
        System.out.println("old speed " + oldSpeed);
    }
}

yourCar = myCar; // copy from one Object to another
yourCar.changeSound("ooga");
yourCar.blowHorn( yourCar.getCurrSpeed()/7 );
myCar.blowHorn(1); // what happens here??

// now ...
// Create a new Car object
// assign it to yourCar
// blow every Car's horn J
```

```

/**
 * all anyone every cares about when abstractig a car J
 */
class Car
{
    private int currSpeed;
    private String hornSound;

    /**
     * Create a new car that is not moving
     * @param - the sound the car makes when you hit the horn
     */
    public Car(String sound)
    {
        currSpeed = 0;
        hornSound = sound;
    }

    /**
     * update the horn sound
     * @param newSound - the improved sound this car now makes
     */
    public void changeSound(String newSound)
    {
        hornSound = newSound;
    }

    /**
     * blow the horn!
     * @param numTimes - the number of times to actualy hit the horn!
     */
    public void blowHorn(int numTimes)
    {
        while (numTimes > 0)
        {
            System.out.println(hornSound);
            numTimes--;
        }
    }

    /**
     * change the cars speed
     * @param milesPerHour - the amount to change the speed by
     */
    public void accelerate(int milesPerHour)
    {
        currSpeed = currSpeed + milesPerHour;
    }

    /**
     * @return - the current speed
     */
    public int getCurrSpeed()
    {
        return currSpeed;
    }
}

```

Exam Review

- OOA – example from class (lemonade, bay bridge, ticket sales, etc.)
- Java Syntax
- Assignments – help a friend, Bill’s parking lot, Rimmer’s Restaurant
- Homeworks – 20.1, 21.7, 21.8, 57.3, 68.1, 77.2, 100.16, 100.19
- Labs – guess, first applet, atom, shape inheritance, code game
- Book – chapters 2,3
- Lectures – every gripping word of it!