**Part 1**  
Given the following set of classes:  
  
interface Domestic { }  
interface Import { }  
interface Japanese extends Import { }  
interface German extends Import { }  
interface Detroit extends Domestic { }  
interface SpringHill extends Domestic { }  
     
interface Vehicle { }  
interface Automobile extends Vehicle { }  
interface LargeAutomobile extends Vehicle { }  
interface Sedan extends Automobile { }  
interface Van extends LargeAutomobile { }  
interface Truck extends LargeAutomobile { }  
interface Compact extends Automobile { }  
interface SportsUtilityVehicle extends Truck, Van { }  
     
class SaturnSL1 implements SpringHill, Sedan { }  
class HondaCivic implements Japanese, Compact { }  
class MercedesC230 implements German, Sedan { }  
class ChevyS10 implements Detroit, Truck { }  
class SubaruOutback implements Japanese, SportsUtilityVehicle { }

For each of the following lines of code, tell me whether the assignment will compile, and if so, if it will succeed or fail at runtime. For any that fail to compile, pretend that the right-hand side of that statement is null, and answer the remaining questions assuming this change has been made. For example, in the following,  
  
 1. Cat c = new Dog( );  
 2. c = new PersianCat( )

assignment number 1 will fail to compile. Answer number 2 assuming that 1 reads  
  
Cat c = null;

|  |  |  |  |
| --- | --- | --- | --- |
| **Table: Statements** | | | |
| Statement | Compile without errors? | Run without errors? | Reason for Error |
| SaturnSL1 sl = new SaturnSL1( ); |  |  |  |
| HondaCivic hc = new HondaCivic( ); |  |  |  |
| Japanese jp = new Japanese( ); |  |  |  |
| German gr = new MercedesC230( ); |  |  |  |
| ChevyS10 cs = new Truck( ); |  |  |  |
| SubaruOutback sb = new SubaruOutback( ); |  |  |  |
| Domestic dm = sl; |  |  |  |
| dm = hc; |  |  |  |
| dm = (Domestic) hc; |  |  |  |
| dm = cs; |  |  |  |
| dm = (Domestic) cs; |  |  |  |
| Import im = sb; |  |  |  |
| gr = im; |  |  |  |
| gr = (German) im; |  |  |  |
| jp = im; |  |  |  |
| jp = (German) im; |  |  |  |
| jp = (Japanese) im; |  |  |  |
| Automobile a = cs; |  |  |  |
| a = hc; |  |  |  |

**Part 2**  
  
Add the following declaration to interface Vehicle:  
  
int getWeightInPounds( );

Then add the minumum amount of additional code to make all the classes that implement Vehicle compile again. Assume that a small car weighs 1,000 pounds, and a large pickup truck weighs 2,500 pounds. Try to write as little redundant code as possible. You may introduce new classes if this would help meet this goal.  
  
Write a class ParkingGarage which simulates a garage for parking Vehicles. The garage should have have enough capacity to park 20 Vehicles or 25,000 pounds. It should refuse to park any Vehicle that would push the garage over either of these limits. Note that ParkingGarage should be a well-behaved class; in particular, it should be possible to create multiple independent ParkingGarages in one program.  
  
ParkingGarage should have a main routine which demonstrates its capabilities. main( ) need not accept any user input.