

**AP[®] COMPUTER SCIENCE A
2006 SCORING GUIDELINES**

Question 2: Taxable Items (Design)

Part A:	<code>purchasePrice</code>	2 1/2 points
----------------	----------------------------	---------------------

- +1 `call getListPrice()`
- +1 calculate correct purchase price (*no penalty if truncate/round to 2 decimal places*)
- +1/2 return calculated price

Part B:	<code>Vehicle</code>	6 1/2 points
----------------	----------------------	---------------------

- +1/2 `class Vehicle extends TaxableItem`
- +1/2 `private double dealerCost`
- +1/2 `private double dealerMarkup` (*no penalty if also store tax in field*)
- +2 1/2 constructor
 - +1/2 `Vehicle(double ?, double ?, double ?)`
 `int/float` (*OK if match fields*)
 - +1 call parent constructor
 - +1/2 attempt using `super`
 - +1/2 correct call: `super(rate)` (*note: must be first line in method*)
 - +1 initialize dealer cost and markup fields
 - +1/2 attempt (must use parameters on RHS or in mutator call)
 - +1/2 correct
- +1 `changeMarkup`
 - +1/2 `public void changeMarkup(double ?)`
 `int/float` (*OK if matches field; no penalty if returns reasonable value*)
 - +1/2 assign parameter to markup field
- +1 1/2 `getListPrice`
 - +1 `public double getListPrice()`
 - +1/2 return sum of dealer cost and markup fields

Note: -1 usage if reimplement `purchasePrice` to do anything other than
 `return super.purchasePrice();`