CS-103

Autumn 2013

Tut 5 (Inheritance)

Q1. Consider a super class as defined below:

class fixDeposit {

protected int accno; // account number

protected double amount; // principal amount

public fixDeposit(int a, double p) {accno = a; amount = p;}

public double interest() { System.out.println(“The instance method “ + “in fixDeposit”) ; return 0;}

public void update(double d) { amount += d; }

public void display() { System.out.println(“The instance method “ + “in fixDeposit”); }

}

Create subclasses (from super class fixDeposit) (i) SIdeposit, and (ii) CIdeposit for accounts earning Simple Interest (SI), and Compound Interest (CI) respectively on the principal amount. Besides inherited members (fields/instance varibles, each subclass should have additional data members for (i) yearly rate of interest, and (ii) time period of deposit in number of years. Define following methods for each subclass:

1. Constructors for the subclasses, explicitly calling superclass constructor
2. The method ‘interest’ which overrides the method ‘interest’ of the superclass, calculates and returns simple/compound interest
3. The function ‘display’ which overrides the method ‘display’ of the superclass, updates the principal amount by adding interest and then displays the final amount

Write a test driver class and explain how the polymorphic feature in Java is implemented in the above.

**Q2.** Modify the program in Q 1 for auto-increment of **account** number of a customer. i.e. if 1 is account no. for 1st customer then 2 should be the account number of the second customer.

**Q3.** Modify the program in Q 1 to make use of ‘abstract**’** class and abstract methods for the same application and explain if any difference in implementation of Q1 and Q2. Are the results same?