ISL38B - OBJECT ORIENTED PROGRAMMING WITH JAVA LAB

LAB TEST – 1 , TERM 28 AUG – 20 DEC 2017

EXTRA QUESTION SET

What will you do to accomodate these scenarios. Write a program with main function to demonstrate this.

1) A base class called 'Bakery' contains a string attribute 'item_name', and two number attributes 'price' and 'qty'. It also has a method called 'priceToPay' that returns the total price to be paid by the user.
Scenairos:
You must not allow the logic of 'priceToPay' to be changed by the sub-classes of 'Bakery'.
Hint: Use 'final' with 'priceToPay' as final int priceToPay( ) { return qty * price; }

2) There is a class called 'Cricket' which contains number attributes 'overs' and 'runs' and a string attribute called 'team_name'. It also has a method called 'display' displaying the details of these attributes.
Scenairo:
You must allow the user at run-time to create as many objects of 'Cricket' as s(he) desires.
Hint: Use 'Array of Cricket objects' as Cricket[ ] objCricket = new Cricket[n] where 'n' value is taken from user at run-time

3) There is a class called 'Animal' which contains attributes such as 'name', 'gender' and 'age'. An unrelated class called 'Vet' contains a string attribute called 'medicine_name' and a member function called 'giveInjection', which displays the name of the injection medicine given to the animal.
Scenairo:
The 'giveInjection' function must be able to give injection to any animal and display the name, gender and age of the animal given the injection.
Hint: Pass an object of animal to 'giveInjection' as 'void giveInjection(Animal a) { System.out.println("Animal "+a.name+" which is "+a.gender+" aged "+a.age+" is injected with "+ medicine_name); }

4) There is a base class called 'Icecream' containing two number attributes 'price' and 'qty', string attributes 'name' and 'flavour'. It also has a method called 'priceToPay' that returns the total price to be paid by the user.
Scenairo:
You must ensure that the sub-class 'ConeIceCream' can be subjected to further derivations while 'CupIceCream' cannot be sub-classed.
Hint: Use final with 'CupIceCream' as final class CupIceCream { //define the class }

5) There is a class called 'Game' which contains string attribute called 'team_name'. It has a sub-class called 'IndoorGame' containing a string attribute called 'game_name'.
Scenairo:
Objects of 'Game' must be able to access the attribute 'game_name' of 'IndoorGame'.
Hint: Use base-class object reference as Game objgame; IndoorGame objig = new IndoorGame( ); objgame = objig; System.out.println("Game name is

"+objgame.game_name);

6) There is a class called 'Jewelery' which contains a string attribute 'jewel_type' and a number attribute 'no_of_grams'.

Scenairos:

a) There need to be an attribute 'jewel_metal' which is set to "gold" and cannot be changed.

b) You must ensure that you can never create object instances of 'Jewelery'.

Hint: Use 'final' with 'jewel_metal' as class Jewelery { final String jewel_metal = "gold"; //rest of class implementation

7) There is a class called 'Books' which contains the string attributes 'title', 'author_name' and 'publisher_name'. The default value of 'publisher_name' is set to "Pearson".

Scenairo:

You must ensure that all objects of 'Books' gets only one shared memory space for the attribute 'publisher_name'

Hint: Use static with 'publisher_name' as class Books { static String publisher_name = "Pearson"; }