WE ARE HAPPY TODAY WITH OOP (Java)

1. Create a class called Book to represent a book. A Book should include four pieces of information as instance variables-a book name, an ISBN number, an author name and a publisher. Your class should have a constructor that initializes the four instance variables. Provide a mutator method and accessor method (query method) for each instance variable. Inaddition, provide a method named getBookInfo that returns the description of the book as a String (the description should include all the information about the book). You should use this keyword in member methods and constructor. Write a test application named BookTest to create an array of object for 30 elements for class Book to demonstrate the class Book's capabilities.

2. (Date Class) Create a class called Date that includes three pieces of information as data members a month (type int), a day (type int) and a year (type int). Your class should have a constructor with three parameters that uses the parameters to initialize the three data members. For the purpose of this exercise, assume that the values provided for the year and day are correct, but ensure that the month value is in the range 1-12; if it is not, set the month to 1. Provide a set and a get function for each data member. Provide a member function displayDate that displays the month, day and year separated by forward slashes (/). Write a test program that demonstrates class Date's capabilities.

3. (Employee Class) Create a class called Employee that includes three pieces of information as data members a first name (type string), a last name (type string) and a monthly salary (type int). Your class should have a constructor that initializes the three data members. Provide a set and a get function for each data member. If the monthly salary is not positive, set it to 0. Write a test program that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10 percent raise and display each Employee's yearly salary again.

4. (Invoice Class) Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as data members a part number (type string), a part description (type string), a quantity of the item being purchased (type int) and a price per item (type int). Your class should have a constructor that initializes the four data members. Provide a set and a get function for each data member. In addition, provide a member function named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as an int value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0. Write a test program that demonstrates class Invoice's capabilities.

5. Create a class called Car that includes three instance variables:

- A model (type String)
- A year (type String)
- A price (double).

Provide a constructor that initializes the three instance variables then provide a set and a get method for each instance variable. If the price is not positive, do not set its values. Write a test application named CarApplication that demonstrates class Car's capabilities. Create two Car objects and display each object's price. Then apply a 5% discount on the price of the first car and a 7% discount on the price of the second. Display each Car's price again.

6. Create a java class called Dog with two instance variables name (typeString) and age (typeInt) then Define a constructor that will initialize these variables and provide a getter and setter methods for both name and age. Finally write a program that demonstrates the creation of a Dog objects, setting its attributes and display them.

7. Create a class named Cat. Declare two (2) instance variables: catType (String) and cutenessLvl (int). Create a public constructor without parameters. Inside the constructor, set the values of catType and cutenessLvlto "Unknown" and 3, respectively. Add another public constructor with two (2) parameters: type (String) and Ivl (int). Inside the constructor, assign type to cat Type and level to cutenessLvl. Create an int type of method named getCuteness(). This method shall return the cat's cuteness level.

8. (Rectangle Class) Create a class Rectangle with attributes length and width. Provide member functions that calculate the perimeter and the area of the rectangle.

9. Create a class called Student with instance variable name (String), age (int) and grade (float) then provide a parameterized constructor to initialize these variables also provide a getter and setter for each instance variable. Additionally include a method called displayInfo() that will print the students name, age and grade. Write a program that create two student objects set their attribute and calls the displayInfo() method for each student.

10. (Rectangle Class) Create a class Rectangle with attributes length and width, each of which defaults to 1. Provide member functions that calculate the perimeter and the area of the rectangle. Also, provide set and get functions for the length and width attributes. The set functions should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0.

10. Consider the following program and then give an output based on the question below: Assume the program is running without an error.

```java
public class Myclass{


       private static int count = 0;
       private int x;

       public Myclass(int i){
               x = i;
       }

        public void incrementCount(){
               count++;
       }

       public void printX(){
               System.out.println("Value of x is: " +x);
       }

       public static void printCount(){
               System.out.println("Value of count is: " +count);
       }



       public static void main(String[]args){

               Myclass myobject1 = new Myclass(5);
               Myclass myobject2 = new Myclass(7);

                     a.  myobject1.printX();
                     b.  myobject1.incrementCount();
                     c.  Myclass.incrementCount();
                     d.  myobject1.printCount();
                     e.  myobject2.printCount();
                     f.  myobject2.printX();
                     g.  myobject1.setX(14);
                     h.  myobject1.incrementCount();
                     i.  myobject1.printX();
                     j.  myobject1.printCount();
                     k.  myobject2.printCount();


   }
}
```