1. Write the code to implement the concept of inheritance for Vehicles. You are required to implement inheritance between classes. You have to write four classes in java i.e. one superclass, two sub classes and one driver class.

Vehicle is the super class whereas Bus and Truck are sub classes of Vehicle class. Transport is a driver class which contains main method.

<u>Detailed description:</u>

Detailed description of Vehicle (Super class):

The class Vehicle must have following attributes:

1. Vehicle model
2. Registration number
3. Vehicle speed (km/hour)
4. Fuel capacity (liters)
5. Fuel consumption (kilo meters/liter)

The Vehicle class must have following methods:

1. Parameterized constructor that will initialize all the datamembers with the given values.
2. Getters and Setters for each data member that will get and setthe values of data members of class.
3. A method ***fuelNeeded()*** that will take***distance (in kilo meter)*** as an argument.It will calculate the amount of fuel needed for the given distanceand will return the value of fuel needed for given distance. Youcan use the attributes '*Fuel consumption'*defined within above Vehicle class to determine the fuel needed forthe given distance. You are required to implement thisfunctionality by yourself.
4. A method ***distanceCovered()*** that willtake ***time*** (in hours) as an argument. Itwill calculate the distance for the given time and speed andreturns the value of distance. The formula to calculate speed isgiven as **speed = distance/time**. You can use thisformula to calculate the distance.
5. A ***display()*** method that will displayall the information of a vehicle.

Detailed description of Truck (Sub class):

The class Truck must have following attribute:

Cargo weight limit (Kilo grams)

The above class must have following methods:

1. Parameterized constructor that will initialize all data memberswith the given values.

2. Getters and setters for each data member that will get and setthe values of data members of class.
3. It must also override the ***display()***method of Vehicle class and must call display() method of superclass within overridden method.

Detailed description of Bus (Sub class):

The class Bus must have following attribute:

No of passengers

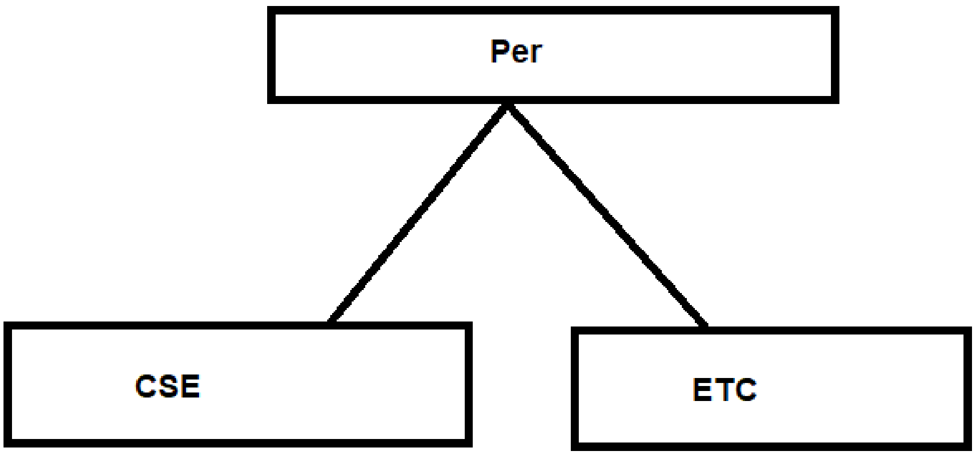
The above class must have following methods:

1. Parameterized constructor that will initialize all the datamembers with given values.
2. Getters and setters that will get and set the value of eachdata member of class.
3. It must also override the ***display()***method of Vehicle class and must call display method of super classwithin overridden method.

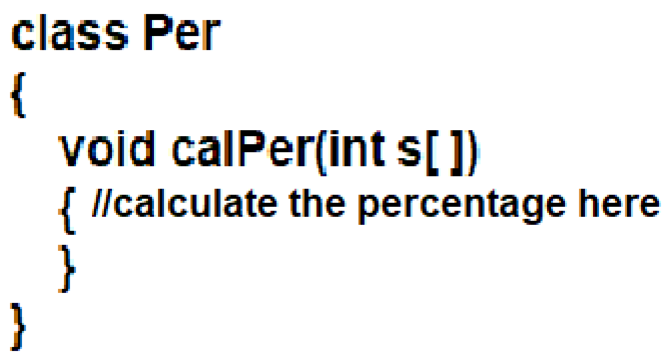
Create a class ***Transport*** whichcontains the main method. Perform the following within mainmethod:

- Create an instance of class Truck and initialize all the datamembers with proper values.
- Create an instance of class Bus and initialize all the datamembers with proper values.
- Now, call **fuelNeeded*()***,***distanceCovered****()* and***display()*** methods using objects of theseclasses.

2. WAP To implement the concept of Inheritence for Engg Result.You are required to implement inheritance between classes. You have to write four classes in java i.e. one superclass, two sub classes and one driver class.



In Percentage class contain the one method name as calPer(int s[])   with subject marks array

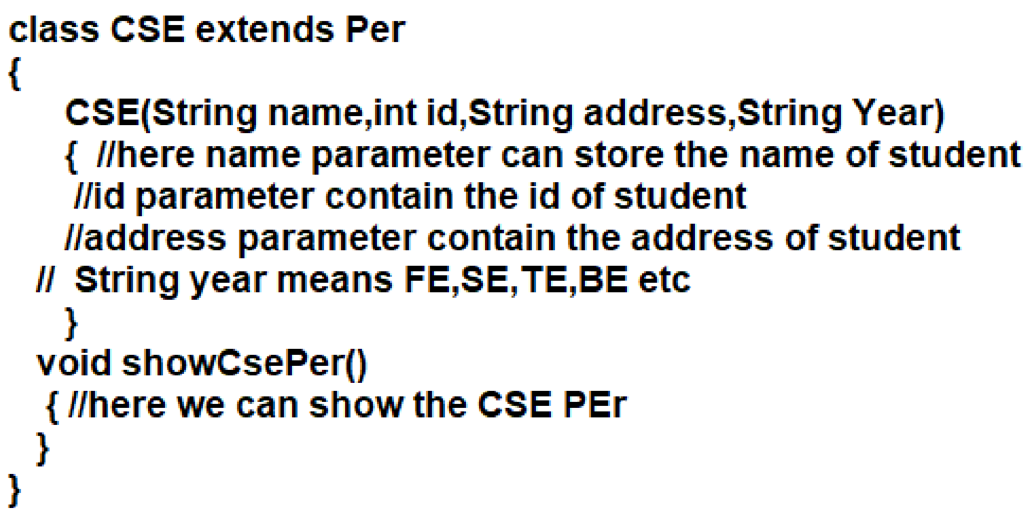


```
class Per
{
   void calPer(int s[ ])
   { //calculate the percentage here
   }
}
```

There are two child classes First is CSE and Second is ETC so first we will discuss about the CSE class

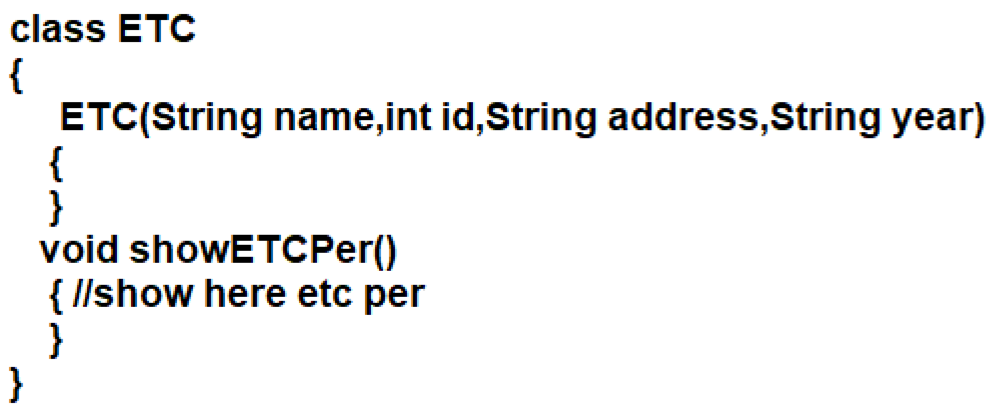
In CSE class contain the parameterized constructor and one method name as showCsePer()

Following sample code



```
class CSE extends Per
{
     CSE(String name,int id,String address,String Year)
     {  //here name parameter can store the name of student
      //id parameter contain the id of student
     //address parameter contain the address of student
  //  String year means FE,SE,TE,BE etc
     }
   void showCsePer()
    { //here we can show the CSE PEr
   }
}
```

In ETC class contain the one parametarized constructor and one method name as showETCPer()

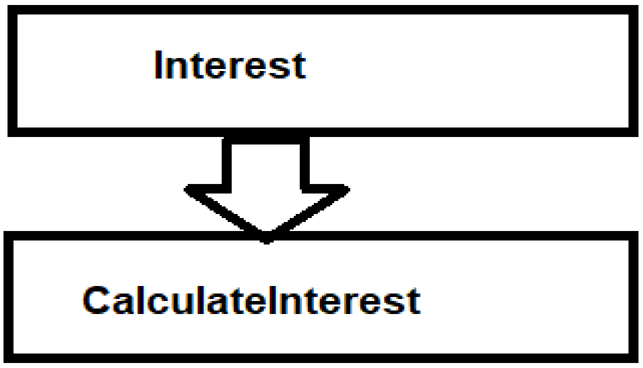


```
class ETC
{
    ETC(String name,int id,String address,String year)
   {
   }
  void showETCPer()
   { //show here etc per
   }
}
```

Then we need to create the main class name as ResultApp and in class we need to write the main method given below

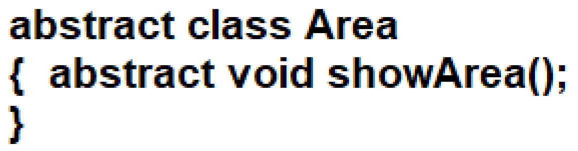
```
class ResultApp
{
   public static void main(String x[])
   {
    //here we need to declare the array with size 6
    //create the object of Scanner class
   //store the values in array

  //create the object of CSE class  and pass the prameter to CSE constructor as per parameter sequence
  //as well as call the  calPer() method of Per and pass array in it
  //call the showCSEPer() method of display the percentage
 //Create the object of ETC class and pass parameter to ETC constructor as per parameter sequence
//call the calPer() method of Per class through ETC class object and pass array in it
 //call the showETCPer() method

   }
}
```

3. WAP using inheritance with a simple Interest Rate Formula for calculate the interest on loan we have the following class hierarchy and conditions given below



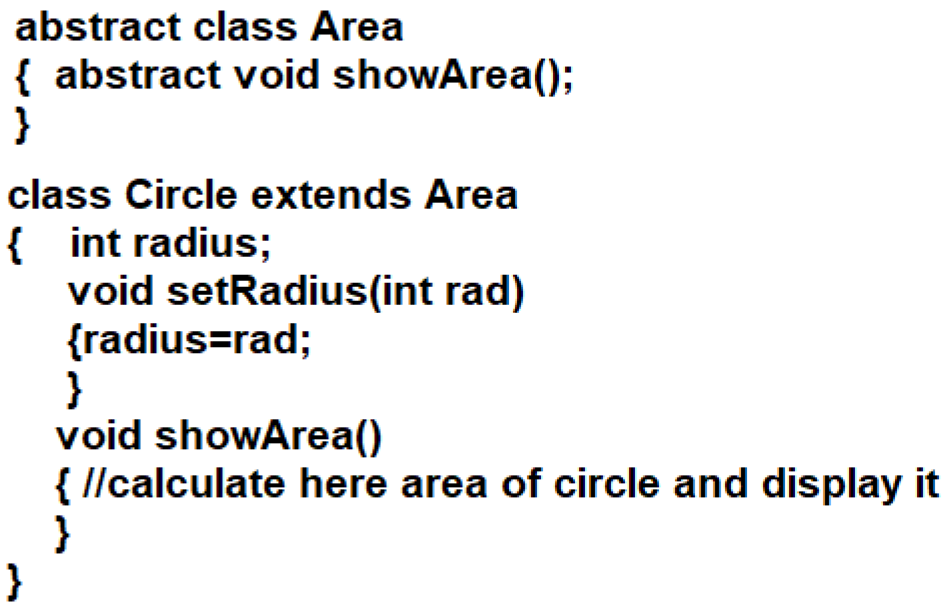
```
class Interest
{   void setPRD(int pamt,int irate,int dur)
     {  //store here principal amount,rate and duration
     }
}
class CalculateInterest extends Interest
{   void calInterest()
     { //here we need to apply the formula and calculate the interest
     }
}
class InterestApp
{
    public static void main(String x[ ])
    { //create the object of Scanner and declare the three variable pamout,rate,dur
     //store the values in pamount,rate,dur
    //create the object of CalculateInterest and call the setPRD and calInterest() function

    }
}
```

4. WAP to create the abstract class name as Area with showArea() abstract method and declare the two child classes name as Circle and Rectangle with a following methods



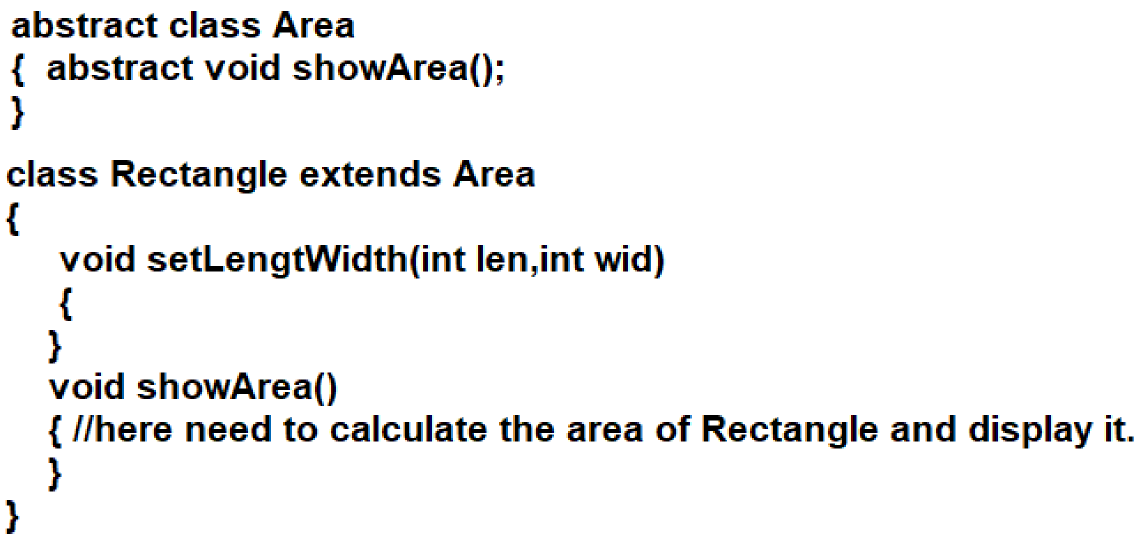
```
abstract class Area
{  abstract void showArea();
}
```

Create the class name as Circle and inherit the Area class in it and override the showArea() as well as define the setRadius() method in Circle class.



```
abstract class Area
{  abstract void showArea();
}

class Circle extends Area
{    int radius;
     void setRadius(int rad)
     {radius=rad;
     }
    void showArea()
    { //calculate here area of circle and display it
    }
}
```

Declare the class name as Rectangle and inherit the Area class in it and override the showArea() method with a setLenghtWidth() function



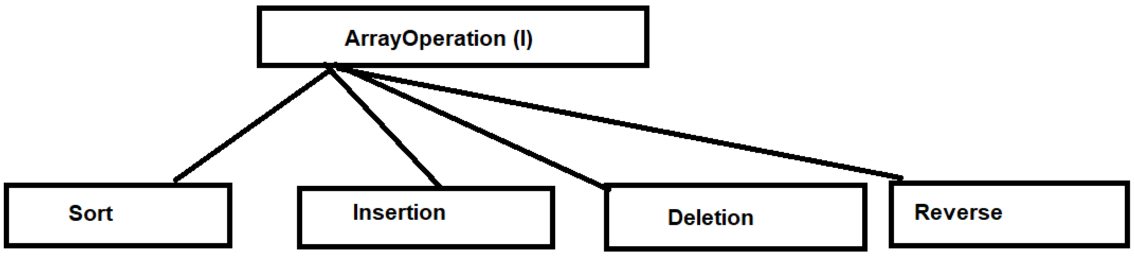
```
abstract class Area
{  abstract void showArea();
}

class Rectangle extends Area
{
     void setLengtWidth(int len,int wid)
      {
    }
    void showArea()
    { //here need to calculate the area of Rectangle and display it.
    }
}
```

Create the main name as AreaApplication

```
class AreaApplication
{
   public static void main(String x[ ])
   {  //create the object of Circle class and call the setRadius() and pass value in it
     //call the object of showArea()
    //create the object of Rectangle class and call the setLengthWidth() method
   //call the showArea() method
   }
}
```

5.WAP create interface name as ArrayOperation with method name as PerformOperation(int [])

And its implementor classes given below



```
interface ArrayOperation
{  void performOperation(int x[ ]);
}
class Sorting implements ArrayOperation
{
     public void performOperation(int x[ ])
     {
        //write here array sorting logics
     }
}
class Insertion implements ArrayOperation
{            int no,ind;
          void setValue(int value,int index)
           { no=value;
             ind=index;
           }
          public void performOperation(int x[])
          {  //perform logics for insert the value on specified index in array
          }
}
```

```
class Deletion implements ArrayOperation
{       int ind;
        void setIndex(int index)
        {  ind=index;
        }
        public void performOperation(int x[ ])
        {   //write here logics to delete the element from array on specified index.
        }
}
```

```
class Rev implements ArrayOperation
{
    public void performOperation(int x[ ])
    {  //write here logics for array reverse
    }
}
```

In above example we define the interface name as ArrayOperation with method performOperation(int[]) meaning is we can perform the operation on array but we cannot mention which operation depending on child implementor classes

Means we can implement the interface in Sorting class and implement the ArrayOperation and write the sorting logics in performOperation() method

In Insertion class implement the Interface ArrayOperation and override the ArrayOperation method in it but before that we need to call the setValue() method in this method can pass the index and value as two parameter for insert the value on specified index

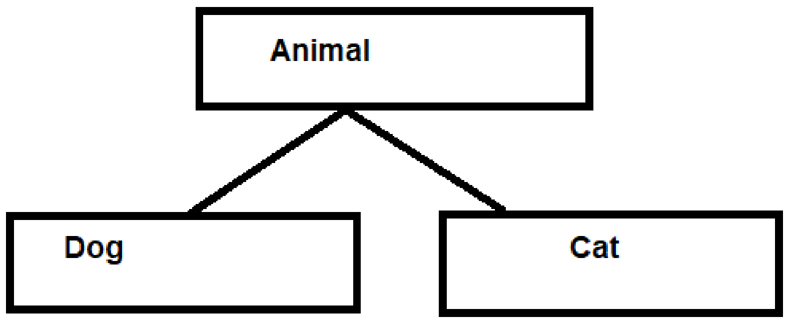
And so on

Lastly we need to create the Main Method class name as ArrayInfOperationApp

When we press the 1 then call the Sorting logics and when we press the 2 then call the Insert value on specified index in array when we press the 3 then call the Deletion element from array logics and when we press 4 then perform array reverse operation

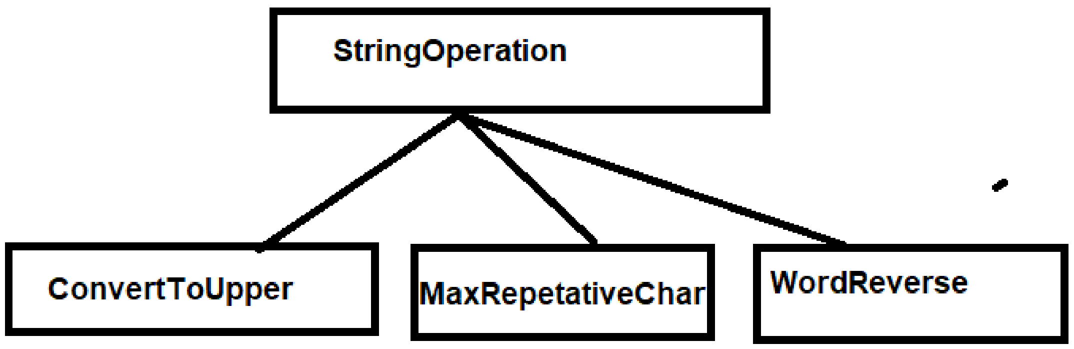
```
public class ArrayInfOperationApp
{
    public static void main(String x[ ])
    {
     //create the object of Scanner class
   //declare the array of six size and store values in it
  //create the variable name as choice and store value in it using Scanner
 //pass the choice variable in switch and perform operation according above mention description


    }
}
```

6.WAP to create the abstract class name as Animal with method makeSound() and its two child classes name as Dog and Cat.We need to inherit the Animal class in Dog and Cat and override the makeSound() method in it and write the message as per sound as his category.



```
abstract class Animal
{  abstract void makeSound();
}
class Dog extends Animal
{   void makeSound()
    {//write here logics as per dog means say bark
    }
}
class Cat extends Animal
{ void makeSound()
   { //write here logics means say mau
   }
}
public class AnimalApp
{ public static void main(String x[])
   {  //here we need to create the object of Dog and Cat and call the makeSound() method

   }
}
```

7.WAP to create the abstract class name as StringOperation with method name as performOperation(String) and write its different logics in different child classes as per give diagram.



Above diagram source code structure given below only you need to put the logics in method.

```
abstract class StringOperation
{
  abstract void performOperation(String x);
}
```

We need to inherit StringOperation class in ConvertToUpper class for convert the string from lower case string to upper case string

Means here we need to create the class name as ConvertToUpper class and override the performOperation() method of StringOperation and write the manual logics for convert lower case string to upper case without using inbuilt function of string.

shown in following code snippets.

```
        abstract class StringOperation
        {
abstract  void performOperation(String);
        }
        class ConvertToUpper extends StringOperation
        {
           public void performOperation(String x)
           {   // here we need to write the manual logic for convert the upper case string to lower case
           }
        }
```

Again we need to inherit the StringOperation class in MaxRepetativeChar class and override the performOperation() method in MaxRepetativeChar class and check which character having a max repetation in string without using a inbuilt function of string.

Shown in following code snippets

```
         abstract class StringOperation
         {
 abstract  void performOperation(String);
         }

        class MaxRepetativeChar extends StringOperation
        {
             void performOperation(String x)
             { //here we need to write the manual logics for count the max reperative char in string.
             }
        }
```

Again we need to inherit the StringOperation class in WordReverse class and override the performOperation() method in WordReverse class and reverse the only word characters in string not reverse the string suppose

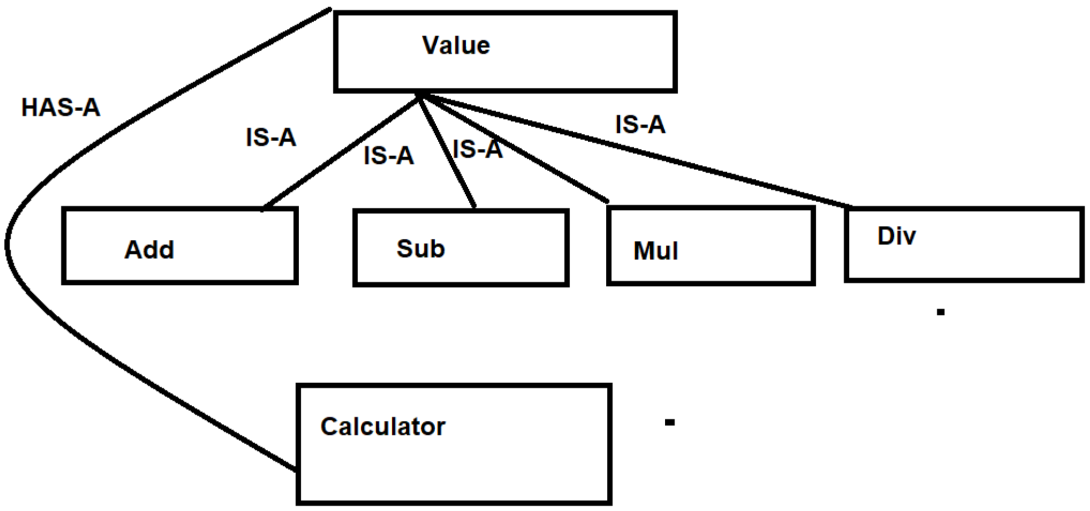
e.g Good Morning India

output like as : dooG gninroM aidnI

Shown in following code snippets

```
          abstract class StringOperation
          {
   abstract  void performOperation(String);
          }


          class WordReverse extends StringOperation
          {
             public void performOperation(String x)
             {
              //here we need to write logics for word reverse in string.
            }
          }
```

8.WAP to design the Calculator Example using a Loose Coupling ?



Explanation of above diagram

In above diagram we have the Value is abstract class and it contain the performCalculation() method

It is abstract method and in this class contain the one another method name as setValue() is normal method here we can accept the two values and store in instance variable shown in given code .

```
abstract class Value
{
   int a,b;
   void setValue(int x,int y)
    { a=x;
      b=y;
    }
  abstract void performCalculation();
}
```

And this class is parent class of Add,Sub ,Mul,Div classes. We need to inherit the Value class

In all child classes and we need to override the performCalculation() method and write the logics

Differently in every child classes shown in following code snippet.

```
abstract class Value
{
   int a,b;
   void setValue(int x,int y)
    { a=x;
      b=y;
    }
  abstract void performCalculation();
}
class Add extends Value
{
     void performCalculation()
     { //write here addition logics
     }
}
class Mul extends Value
{
    void performCalculation()
    { //write here multiplication logics
    }
}

   class Div extends Value
   {
       void performCalculation()
       { //write here division logics
       }
   }
   class Sub extends Value
   {
       void performCalculation()
       { //write here substraction logics
       }
   }
```

We have another class name as Calculator this class is depend on Value class for perform any operation like as addition,substraction,multiplication or division

Means for this purpose we create method name as performOperation(Value) s

In this method we pass the reference of Value class and it is parent class of all classes means this method can accept the reference of any of its child classes and call performCalculation() as per the child object shown in following code.

```
class Calculator
{
    void performOperation(Value v)
    {
        v.performCalculation();
    }
}
```

Again we need to create the main method class and and perform the following operation as per given instruction

```
public class LooseCouplingApp
{
  public static void main(String x[])
  {  //create the object of Scanner class and declare the variable as choice
     // ask programmer to enter your choice and enter the choice from keyboard using scanner
    // create the object of Calculator class
   // write here code for switch statement
  // in case 1 create the object of Add class and call setValue() method using Add object and pass two values in it
 //call the performOperation() method of Calculator class and pass reference of Add class in it.
//in case 2 create the object of Mul class and call the setValue() method using Add() object and pass two value in it
// call the performOperation() method of Calculator() and pass referece of Mul class in it
 //perform the case 3 and 4 like case 1 and 2 with Div and Sub classes

  }
}
```