Immutable class

1. Class must be declared final
2. Data members must be declared final
3. A parameterized constructor
4. Getter methods
   1. In case of date return new reference of date
   2. In case of list return new reference of list or return a reference of object by using index.
5. No setter methds

package org.immutable.pojo;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

public final class **Student** {

private final String name;

private final int age;

private final Date dob;

private final List<Subject> marks;

public Student(String name, int age, Date dob, List<Subject> marks) {

super();

this.name = name;

this.age = age;

this.dob = dob;

this.marks = marks;

}

public String getName() {

return name;

}

public int getAge() {

return age;

}

//make date read only

public Date getDob() {

return new Date(dob.getTime());

}

//Make arraylist read only

public List<Subject> getMarks() {

return new ArrayList<Subject>(marks);

}

@Override

public String toString() {

return "Student [name=" + name + ", age=" + age + ", dob=" + dob + ", marks=" + marks + "]";

}

}

**package** org.immutable.pojo;

**public** **final** **class** Subject {

**private** **final** String subName;

**private** **final** **int** marks;

**public** Subject(String subName, **int** marks) {

**super**();

**this**.subName = subName;

**this**.marks = marks;

}

**public** String getSubName() {

**return** subName;

}

**public** **int** getMarks() {

**return** marks;

}

@Override

**public** String toString() {

**return** "Subject [subName=" + subName + ", marks=" + marks + "]";

}

}

package com.demo.spring;

import java.util.Date;

import java.util.LinkedList;

import java.util.List;

import org.immutable.pojo.Student;

import org.immutable.pojo.Subject;

/\*\*

\* Hello world!

\*

\*/

public class App

{

public static void main( String[] args )

{

Subject s1 = new Subject("Maths", 100);

Subject s2 = new Subject("English", 81);

Subject s3 = new Subject("Science", 79);

List<Subject> l = new LinkedList();

l.add(s1);

l.add(s2);

l.add(s3);

Student s = new Student("Abrar", 22, new Date("1993/01/22"), l);

Subject s4 = new Subject("History", 50);

System.out.println("s1111"+s.getMarks());

System.out.println("size before adding subject :: "+s.getMarks().size());

System.out.println("adding s4 :: "+s4);

s.getMarks().add(s4);

System.out.println("size after adding subject :: "+s.getMarks().size());

// Changing date field

System.out.println("date of birth before change ::"+s.getDob());

s.getDob().setDate(12);

System.out.println("date of after before change ::"+s.getDob());

System.out.println(s);

}

}

Output :::

s1111[Subject [subName=Maths, marks=100], Subject [subName=English, marks=81], Subject [subName=Science, marks=79]]

size before adding subject :: 3

adding s4 :: Subject [subName=History, marks=50]

size after adding subject :: 3

date of birth before change ::Fri Jan 22 00:00:00 IST 1993

date of after before change ::Fri Jan 22 00:00:00 IST 1993

Student [name=Abrar, age=22, dob=Fri Jan 22 00:00:00 IST 1993, marks=[Subject [subName=Maths, marks=100], Subject [subName=English, marks=81], Subject [subName=Science, marks=79]]]