CS252 (2017-18-I)

Lab Exercise 1

Title: A warm-up to Java Programming

Objective: To test basic understanding of the Java language paradigms

Problem Set – Wednesday, 9th August, 2017

Problem 1: Core language
[10 Marks]

a. Declare an *abstract* class called `Video`, with the following properties:
   i. Play duration – A small positive real number
   ii. Release Date – Date [Hint: See details of the class *java.util.Date*]

   [1 Mark]

b. Add a constructor to initialize the above properties. Declare *getter* and *setter* methods for both properties. Provide concrete implementation for the getter and setter method pertaining to Play duration, while leaving those pertaining to Release date, to be implemented by any subclass.
   [2 Mark]
c. Declare a class called `Movie`, which extends `Video`. Add the following properties to the class:
   i. Movie name – A sequence of words
   ii. Rating – "A", "U" or "UA". [Hint: See *Enum* data types]

   [2 Marks]

d. Add a constructor to initialize the above properties. Add getter and setter methods for the newly added properties.
   [2 Mark]
e. Define a *Natural Object Ordering* for the objects of the class `Movie`, such that:
   i. Two objects are *equal*, if the *name of the movie* as well as the *release date* are the same.
   ii. Otherwise the object, in which the *name of the movie* is lexicographically smaller, is considered smaller in the ordering.
   iii. For objects in which the *name of the movie* is same, but the *release date* is different, the object with an earlier release date is considered smaller in the ordering.

[Hint: See details of the *Comparable* interface]
[3 Marks]

Problem 2: The Reflection API
[10 Marks]

a. Declare an *abstract* class called `ExamCentres`, with the following *static constants*:
   i. LKO – A small integer with value 10
   ii. CNB – A small integer with value 20
   iii. AGC – A small integer with value 30
   iv. ALD – A small integer with value 40
   v. VNS – A small integer with value 50

   [2 Marks]

b. Declare a class called `Exam`, with the following properties:
   i. Title – A sequence of words
   ii. Centre code – A small integer

   [1 Mark]

c. Add a constructor to the class `Exam` to initialize all the fields. Add getter and setter methods for all the properties.
   [2 Marks]
d. Add a method called `printCentreName()` to the class `Exam`, which prints the name of the Centre, corresponding to the Centre code. For example, if the Centre code is **10**, it should print **LKO**. Similarly, if the Centre code is **50**, it should print **VNS**. If the code corresponding to the Centre code doesn't have a matching, it should print **Unknown**.
   [2 Mark]
e. Add two methods in the `Exam` class:
   i. `void printCentreName()` – prints the name of the Centre, corresponding to the Centre code. For example, if the Centre code is **10**, it should print **LKO**. Similarly, if the Centre code is **50**, it should print **VNS**. If the code corresponding to the Centre code doesn't have a matching constant in the class `ExamCentres`, it should print **Unknown**.
   ii. `boolean isValidCentre(int centreCode)` – returns **true**, if the passed Centre code, has a corresponding constant defined in the class `ExamCentres`, returns **false** otherwise.

   Keep in mind that new constants can be added in the class `ExamCentres`, independently of any changes in the class `Exam`.
   [Hint: You can iterate over *all the members* defined in a class, with specific set of

modifiers]

[(2 + 3) Marks]