



University  
of Windsor

SCHOOL OF COMPUTER SCIENCE  
03-60-212 – OOP USING JAVA  
SUMMER 2014

LAB – 2

[To get the **full marks**, complete and show your works to the Lab Instructor before the end of the Lab period. Lab works submitted in the next week's Lab will get a maximum of **75% marks**, if you attend this Lab, and a maximum of **50% marks**, if you do not attend this Lab.]

**Objective:**

The objective of this Lab is to develop simple Java Classes.

**Part 1:**

Define a class called **Name** with the following members:

- Three private String type instance variables namely, **firstName**, **middleName** and **lastName**, to store the first name, the middle name and the last name of a person, respectively. Note that some people may not have any middle name. For such cases the value of **middleName** should be **null**. For instance **"John Henry Smith"** and **"Mary Todd"** are both valid names.
  - A constructor that takes the full name of a person as its argument of the String class. You must use **StringTokenizer** to extract different components of the name. No matter how the name string is entered, each name should be store such that the first letter of the name is always a capital letter and all other letters are small letters. For example **"John"**, **"john"**, **JOHn** and **"JOHN"**, all should be stored as **"John"**.
  - A copy constructor that takes an object of class **Name** and creates a new object with the same values for **firstName**, **middleName** and **lastName**.
  - Public **accessor** and **mutator (getter/setter)** methods for each attribute. The mutator method must also format the letters of a name as described above.
  - A **toString()** method that returns an object of the String class in the following format:
    - If there is a middle name:
      - Last Name + a comma + a space + first name + a space + the middle initial followed by a period.
    - If there is no middle name:
      - Last Name + a comma + a space + first name
- Example **"Smith, John H."** and **"Todd, Mary"** will correspond to **"John Henry Smith"** and **"Mary Todd"**, respectively.

**Part 2:**

Define another class called **Date** with the following members:

- Three private integer attributes (**day**, **month** and **year**)
- A no-argument constructor (initializing with an arbitrary date as "1/1/1000")
- A constructor with a String argument in the format of "dd/mm/yyyy". Use **StringTokenizer** class to parse the argument passed to the constructor.

- A copy constructor that takes an object of class **Date** and creates a new object with the same values for **day**, **month** and **year**.
- Public **accessor** and **mutator (getter/setter)** methods for each attribute. In the mutator methods, the validity of the arguments should be checked using some private methods (such as, **isValidDay** (1 – 31), **isValidMonth** (1 – 12) and **isValidYear** (1000 - 3000)), before updating the field values.
- A public method **earlierThan**, that takes an object of class **Date** as its parameter. If the date in the current object corresponds to a date **earlier** than that represented by the argument, the method should return **true**. Otherwise, it should return **false**.
- A public method **equals** that takes an object of class **Date** as its parameter. The method should return **true** if the argument corresponds to the same day, month and year as of the current object. Otherwise it should return **false**.
- A public method **toString** that will return a string representing the **Date** object in the following format:
  - Name of the month + a space + day + a comma + a space + year
  - For example the date 18/6/2014 will be printed as "June 18, 2014"
  - NOTE: You may have to use a String array of months.

Test both **Name** and **Date** classes with the test application (**TestNameAndDate**) that is provided. The outputs from your programs should match the sample outputs provided below.

#### Sample output from TestNameAndDate:

Testing Name class...

Todd, Mary

Smith, John H.

Rahman, Quazi R.

Testing Date class...

date1 = July 12, 2008

date2 = November 20, 2010

date3 = September 6, 2002

date1 = July 12, 2008

date2 = November 20, 2010

date3 = September 6, 2002

date1 is earlier than date2

date1 is later than date3

date2 is later than date1

date2 is later than date3

date3 is earlier than date1

date3 is earlier than date2

date4 = January 1, 1000