Date and Time

Locale class

- A Locale object represents a specific geographical, political, or cultural region.
- An operation that requires a Locale to perform its task is called locale-sensitive and uses the Locale to tailor information for the user.
- For example, displaying a Date is a locale-sensitive operation— the Date should be formatted according to the customs and conventions of the user's native country, region, or culture.

Locale Class

- The Locale class is used for worldwide denomination
  - DateFormat class can use an instance of Locale to customize formatted o

Introduction to Date (1/2)

Java API provides an extensive set of classes to work with dates, numbers and currencies

Few date related classes

- **java.util.Date**
  - This class is used to bridge between the Calendar and DateFormat class
  - An instance of Date represents a mutable date and time, to a millisecond

- **java.util.Calendar**

  Provides a huge variety of methods that help to convert and manipulate dates and times

- **java.text.DateFormat**
  - Used to format dates in various styles such as "01/01/70" or "January 1, 1970“

Introduction to Date (2/2)

- **java.text.NumberFormat**

  Used to format numbers and currencies for numerous locales

- **java.util.Locale**
  - Used in conjunction with DateFormat to format dates for specific locales
  - Used in conjunction with NumberFormat to format numbers and currency for specific locales

Date

- Class Date represents a specific instant in time, with millisecond precision
- The no-arg constructor of date class returns the current date and time
- The methods of date can be used to display, manipulate the date and time

Methods

- after(Date d)
  - Tests if this date is after the specified date
- before(Date d)
  - Tests if this date is before the specified date
- getTime()
  - Returns the number of milliseconds since January 1, 1970
- setTime(long time)
  - Sets this Date object to represent number of milliseconds after January 1, 1970

The Calendar Class

- Calendar is an **abstract** class that has many fields and methods to display and manipulate calendar instance

```
Calendar ca = new Calendar();
              //illegal,Calendar is abstract
```

- In order to create a Calendar instance, use one of the overloaded getInstance() static methods

```
Calendar ca = Calendar.getInstance();
```

- Calendar reference variable (ca) is referring to an instance of a concrete subclass of Calendar, java.util.GregorianCalendar

DateFormat Class

- DateFormat is an abstract class
- It belongs to java.text.DateFormat Package
- Use two factory methods, getInstance() and getDateInstance() to format dates
- The format() method is used to create Strings representing the properly formatted versions of the Date
- The static fields like SHORT,MEDIUM,LONG,FULL represents a formatting style

Some more DateFormat method

parse() Method

- Takes a String formatted in the style of the DateFormat instance being used, and converts the String into a Date object

- parse() can throw a ParseException