

CS385 Assignment I (15 points)

Your first programming assignment will be for the purpose of reviewing basic Java. Your prior knowledge of Java and the review lectures of module 1 should be sufficient for doing this assignment. In this assignment you are to write classes for a *simple calendar* application. More specifically, you should write the following three classes:

- **Date.java:** for representing the dates/days
- **Event.java:** for representing an event.
- **Calendar.java:** for storing a set of events

The specification for each class is as follows:

Class Date

Class Definition:

Class Date should implement the “Comparable” interface:

```
public class Date implements Comparable<Date> {  
    //Data fields and Method}
```

Data Fields:

Class Date has three data fields:

- **int year** (must be between 2014-2020)
- **int month** (must be between 1-12)
- **int day** (must be between 1-31)

Constructors:

Class Date has one constructor which initializes year, month, and year. The signature for this constructor is as follows:

```
public Date(int year, int month, int day) throws IllegalArgumentException {  
    //constructor body}
```

You should check to see if each argument is within the specified range and throw an `IllegalArgumentException` otherwise. (If you don't remember the exception handling in java watch this complete tutorial on youtube:

<http://www.youtube.com/watch?v=-aQeyTGqoQc>)

Methods

- Accessors for all three data fields: **int getYear(), int getMonth(), int getDay()**
- **String toString():** produces a string version of the date which should be in the form “month/day/year”
- **Boolean equals(Object obj):** This method overrides the “equals” method in the Object class. It returns true if this Date object has the same year, month and day as obj. Otherwise, it returns false. Before checking for equality “obj” must be casted to “Date as follows:

```
public Boolean equals(Object obj){  
    Date otherDate = (Date)obj;  
    //the rest of the method body}
```

Now you can compare this Date object with “otherDate” object.

- **public int compareTo(Date otherDate):** This method is inherited from Comparable interface and must return
 - 0 if this Date object is equal to the otherDate object
 - -1 if this Date object is earlier than the otherDate object
 - +1 if this Date object is later than the otherDate object

Class Event

Class Definition:

```
public class Event {
//Date fields and Methods}
```

Data Fields:

- **Date date:** (the date of the event)
- **int start:** (The start time/hour of the event. It must be a number between 0-23)
- **int day:** (the end time/hour of the event. It must be a number between 0-23)
- **String description:** (A description of the event.)

Constructors:

Class Event has one constructor which initializes Date, start, end, and description. The signature for this constructor is as follows:

```
public Event (Date date, int start, int end, String description) throws
IllegalArgumentException{
//constructor body}
```

You should check to see if “start” hour is less than or equal “end” hour and throw an IllegalArgumentException otherwise.

Methods:

- Accessors for all three fields: **Date getDate(), int getStart(), int getEnd(), String getDescription().**
- A Mutator for the description field: **void setDescription(String newDescription)**
- **String toString():** Produces a string version of an Event which should be of the form: "month/day/year start--end: description".
- **Boolean equals(Object obj)** – Determines whether two Event objects represent the same event. True if both event objects have the same date, start, end, and description attributes (use equals to compare dates and description).

Class Calendar

```
public class Calendar {
//Date fields and Methods}
```

Data Fields:

- **static final int MAXEVENTS=4 :** A constant representing the maximum number of events that can be stored. Make it just 4 to support easy testing!
- **Event[] events:** An array holding the scheduled events, of size MAXEVENTS. If you don't remember how to initialize an array and access its elements, please refer to this Oracle Tutorial (<http://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>)

- **int numEvents** – The number of events currently scheduled (i.e., the number of entries in the “events” array).

Constructors:

Class Calendar has one default constructor which creates an empty array of events of size MAXEVENTS, and initializes numEvents to zero.

```
public Calendar() {
    //Constructor Body
}
```

Methods:

- **Boolean addEvent(Event e):** Adds an event to the events array if array is not full (numEvents==MAXEVENTS). It returns true if the event is successfully added and false otherwise. If the array is not already full, you should traverse the array and insert the event in place of the first null entry. Don't forget to increment the “numEvents” if an event is added successfully.
- **int findEvent(Event e):** traverses the “events” array to look for an event that is equal to “e”. If such event is found, then the index of it in the array is returned. Otherwise, “-1” is returned. Note that the array may contain null entries so before checking for equality, make sure that the array entry is not null otherwise your program may throw a NullPointerException.
- **Boolean removeEvent(Event e):** removes an event from the array and returns true if the remove operation is successful and false otherwise. Your method should first call FindEvent to retrieve the index of the array entry that contains the event. If FindEvent returns -1 then the event does not exist in the array and your method should return false. Otherwise, your method should set the array entry to null.
- **Void dump():** Prints all the events in the calendar (i.e. all **non-null** entries in the “events” array). Each event should be printed in a separate line.

After you are done with writing the above three classes. Use that attached “**CalendarTest.java**” class to test your three classes. This is the test harness I use for grading your program. Make sure that your classes pass all the tests.

What you should turn in:

You should turn in three java files: “**Date.java**”, “**Event.java**”, and “**Calendar.java**”. Please do not leave extra/unwanted code (particularly print/debug statements) in your submission.

Please do not hesitate to email me if you have any questions.

Grading Rubric:

Classes , data fields, and accessor/mutator methods are properly defined	3 pt
Constructors and other methods pass all the tests	12 pt
Total	15pt