

## How to add, subtract days, months, years, hours from Date and Time in Java

Adding days, hours, month or years to dates is a

common task in Java. `java.util.Calendar` can be

used to perform Date and Time arithmetic in Java.

`Calendar` class not only provides date manipulation but

it also support time manipulation i.e. you can add,

subtract hours, minutes and seconds from current time.

`Calendar` class automatically handles date transition or

month transition for example if you ask date after 30

days it will return you date based on whether current

month is 30 or 31 days long. Same is true in case of

adding and subtracting years, `Calendar` takes care

whether current or following year is a [leap year](#) or not.

For example 2012 is a leap year and it has February with 29 days, if you ask `Calendar` day before 365 it will return

24th July (assuming current date 23<sup>rd</sup> July) which shows it take care of leap year. By the way there are couple of more date and time related articles e.g. [How to find current date and time in Java](#) and [How to convert Date to String in Java](#).

If you haven't read them already, It's worth checking to know more about Date and Time in Java.

## How to add or subtract days, month and year from date in Java



`Calendar.DATE` field can be used to add or subtract [dates in Java](#). Positive value passed

into `add()` method will **add days** into date while negative values will subtract days from date in Java.

Similarly `Calendar.MONTH` can be used to add and subtract months from date in Java. You can use this to get date after 2 months or before 2 months. Just keep in mind that `Calendar.MONTH` start from zero.

Same `add()` method can be used to both add or subtract [months in Java](#). Once again difference will only be on

whether value is positive or negative. `Calendar.YEAR` can be used to add or subtract year from current date in the same fashion we added days and month into date.

## How to add or subtract hour, minutes and seconds from Time in Java

Calendar class not only provides Date information but also gives Time related information. You can get hours, minutes and seconds from `java.util.Calendar` instance. Similarly you can use same `add()` method for adding or subtracting hours, minutes and seconds from current time in Java. Just need to careful whether you are using `Calendar.HOUR` or `Calendar.HOUR_OF_DAY` because former represent time in AM and PM while later represent time in 24 hours time. `Calendar.MINUTES` can be used for adding or subtracting minutes from Date.

## Important points related to Calendar class in Java

- 1) Calendar has [overloaded](#) `getInstance()` method which can return Calendar instance in either default timezone and locale or specified timezone or locale.
- 2) Calendar has different fields to return specifics from Calendar like `Calendar.DATE`, `Calendar.MONTH`, `Calendar.YEAR` etc you can check Javadoc for full list.
- 3) `Calendar.MONTH` return zero for first month, keep in mind while using value returned by `Calendar.get(Calendar.MONTH)`
- 4) `Calendar.HOUR_OF_DAY` represent time in 24 hours format. Calendar also support AM or PM format.

```
import java.util.Calendar;
import java.util.TimeZone;

/**
 * Java program to add, subtract dates, month and year using Calendar in Java.
 * Apart from date, Calendar class also provide time related information and can
 * be used to add and subtract hours, minutes and seconds from time in Java.
```

```
*  
* @author Javin Paul  
*/  
  
public class DateAndTimeArithmetic {  
  
    public static void main(String args[]){  
  
        //Java calendar in default timezone and default locale  
        Calendar cal = Calendar.getInstance();  
        cal.setTimeZone(TimeZone.getTimeZone("GMT"));  
  
        System.out.println("current date: " + getDate(cal));  
  
        //adding days into Date in Java  
        cal.add(Calendar.DATE, 2);  
        System.out.println("date after 2 days : " + getDate(cal));  
  
        //subtracting days from Date in Java  
        cal.add(Calendar.DATE, -2);  
        System.out.println("date before 2 days : " + getDate(cal));  
  
        //adding months into Date  
        cal.add(Calendar.MONTH, 5);  
        System.out.println("date after 5 months : " + getDate(cal));  
  
        //subtracting months from Date  
        cal.add(Calendar.MONTH, -5);  
        System.out.println("date before 5 months : " + getDate(cal));  
  
        //adding year into Date  
        cal.add(Calendar.YEAR, 5);  
        System.out.println("date after 5 years : " + getDate(cal));  
  
        //subtracting year from Date  
        cal.add(Calendar.YEAR, -5);  
        System.out.println("date before 5 years : " + getDate(cal));  
  
        //date after 200 days from now, takes care of how many days are in month  
        //for years calendar takes care of leap year as well  
        cal.add(Calendar.DATE, 200);  
        System.out.println("date after 200 days from today : " + getDate(cal));  
    }  
}
```

```

    System.out.println("current time in GMT: " + getTime(cal));

    //adding hours into Date
    cal.add(Calendar.HOUR_OF_DAY, 3);
    System.out.println("Time after 3 hours : " + getTime(cal));

    //subtracting hours from Date time
    cal.add(Calendar.HOUR_OF_DAY, -3);
    System.out.println("Time before 3 hours : " + getTime(cal));

    //adding minutes into Date time
    cal.add(Calendar.MINUTE, 3);
    System.out.println("Time after 3 minutes : " + getTime(cal));

    //subtracting minutes from Date time
    cal.add(Calendar.HOUR_OF_DAY, -3);
    System.out.println("Time before 3 minuets : " + getTime(cal));

}

/**
 *
 * @return current Date from Calendar in dd/MM/yyyy format
 * adding 1 into month because Calendar month starts from zero
 */
public static String getDate(Calendar cal){
    return "" + cal.get(Calendar.DATE) + "/" +
        (cal.get(Calendar.MONTH)+1) + "/" + cal.get(Calendar.YEAR);
}

/**
 *
 * @return current Date from Calendar in HH:mm:ss format
 *
 * adding 1 into month because Calendar month starts from zero
 */
public static String getTime(Calendar cal){
    return "" + cal.get(Calendar.HOUR_OF_DAY) + ":" +
        (cal.get(Calendar.MINUTE)) + ":" + cal.get(Calendar.SECOND);
}
}

```

**Output :**

```
current date: 23/7/2012
date after 2 days : 25/7/2012
date before 2 days : 23/7/2012
date after 5 months : 23/12/2012
date before 5 months : 23/7/2012
date after 5 years : 23/7/2017
date before 5 years : 23/7/2012
date after 200 days from today : 8/2/2013
current time in GMT: 6:12:53
Time after 3 hours : 9:12:53
Time before 3 hours : 6:12:53
Time after 3 minutes : 6:15:53
Time before 3 minuets : 3:15:53
```

That's all on How to add days, month and year on Date in Java. We have also seen how to add hours, minutes and seconds into time using `java.util.Calendar` class. Two points which is worth remembering is that month starts from zero and time can be represented in either 24 hours format or AM-PM format.

## Further Learning

[Complete Java Masterclass](#)

[Java Fundamentals: The Java Language](#)

[Java In-Depth: Become a Complete Java Engineer!](#)

Other **Date and Time tutorials** from Javarevisited Blog

[Difference between `java.util.Date` and `java.sql.Date` in Java](#)

[How to create Date from String in Java](#)

[SimpleDateFormat is not thread-safe, Use carefully.](#)

[Difference between `java.sql.Time`, `java.sql.Timestamp` and `java.util.Date` in Java](#)

[How to make SimpleDateFormat thread-safe using `ThreadLocal` in Java](#)

00:09 / 00:55

By javin paul at [December 29, 2017](#) 

Labels: [core java](#), [date and time tutorial](#), [programming](#)

Location: [United States](#)

## 2 comments :

**Anonymous said...**

Sorry, you had a mistake here, you put hour instead of minutes

```
//subtracting minutes from Date time  
cal.add(Calendar.HOUR_OF_DAY, -3);  
System.out.println("Time before 3 minuets : " + getTime(cal));
```

Thank's gor your post,  
borchvm