

Lab Exercise: java.date





About Intertech

Thank you for choosing Intertech for your training. The next page of this document will get you started with your lab if you'd like to skip ahead. Below is a brief overview of Intertech as well as a promo code for you for future live Intertech trainings.

Intertech (www.intertech.com) is the largest combined software developer **training** and **consulting** firm in Minnesota. Our unique blend of training, consulting, and mentoring has empowered technology teams in small businesses, Fortune 500 corporations and government agencies since 1991.

Our training organization offers live in-classroom and online deliveries, private on-site deliveries, and on-demand options. We cover a broad spectrum of .NET, Java, Agile/Scrum, Web Development, and Mobile technologies. See more information on our training and search for courses by [clicking here](#).

We appreciate you choosing Intertech!

Lab Exercise

java.date

The new java.date package in Java SE 8 provides an easy to use, and immutable library for handling dates and times.

In this lab, you will challenge yourself to create a lambda which relies on the LocalDate methods to determine if an order is a “priority order.”

Specifically, in this lab you will:

- Create a lambda
- Instantiate LocalDate objects
- Use LocalDate methods to compare one to another.

Scenario

The store has been a massive success. In fact, the average order amount this month has been over \$3,000! Unfortunately, that also means that there were more “Priority Orders” than expected, which, in turn, has caused some customers orders to be delayed for over a month! To get things back on track, you’ve been tasked to temporarily change the priority order criteria: if one month (or more) has placed since an order was created, it should be treated as a priority.

Step 1: On Your Own: Change the lambda implementation to work with LocalDate to determine if an order has been sitting around for a month or longer (but leave the signature of the isRushable() method alone).

If it has been sitting around for a month or longer, return true. Otherwise, return false.

Step 2: Inside of TestOrders.java, create a new “hammer” order with a date that falls within 15 days from today.

```
// Change this date to one that is within the last 15 days of
today.
MyDate hammerDate = new MyDate( 5, 17, 2016 );
Solid hammerType = new Solid( "Acme Hammer", 281, 0.3,
    UnitOfMeasureType.CUBIC_METER, false, 100, 0.25, 0.3 );
Order hammer = new Order( hammerDate, 10.00, "Wile E Coyote",
    hammerType, 10 );
```

Step 3: Verify that the anvil is considered a priority item and the hammer is not.

```
System.out.println("Anvil isPriorityOrder: " +
    anvil.isPriorityOrder());
System.out.println("Hammer isPriorityOrder: " +
    hammer.isPriorityOrder());Lab Solutions
```

Lab Solutions

TestOrders.java

```
package com.acme.testing;

import com.acme.domain.Order;
import com.acme.domain.Service;
import com.acme.domain.Solid;
import com.acme.domain.Good.UnitOfMeasureType;
import com.acme.utils.MyDate;

public class TestOrders {

    public static void main(String[] args) {
        MyDate date1 = new MyDate(1, 20, 2008);
        Solid s1 = new Solid("Acme Anvil", 1668, 0.3,
            UnitOfMeasureType.CUBIC_METER, false, 500, 0.25, 0.3);
        Order anvil = new Order(date1, 2000.00, "Wile E Coyote", s1,
10);

        MyDate date2 = new MyDate(4, 10, 2008);
        Solid s2 = new Solid("Acme Balloon", 1401, 15,
            UnitOfMeasureType.CUBIC_FEET, false, 10, 5, 5);

        // Change this date to one that is within
        // the last 15 days of today.
        MyDate hammerDate = new MyDate( 5, 17, 2016 );
        Solid hammerType = new Solid( "Acme Hammer", 281, 0.3,
            UnitOfMeasureType.CUBIC_METER, false, 100, 0.25, 0.3 );
        Order hammer = new Order( hammerDate, 10.00, "Wile E Coyote",
            hammerType, 10 );

        Order.setRushable( (orderDate, orderAmount) -> {
            // Create a LocalDate object for now.
            LocalDate now = LocalDate.now();
            // Create a LocalDate object for the order date.
            LocalDate orderDatePlus30 =
LocalDate.of(orderDate.getYear(),
                orderDate.getMonth(), orderDate.getDay());
            // Add one month to the order date.
            orderDatePlus30 = orderDatePlus30.plusMonths(1);
            // If the current date is after the order date + one
month,
            // it's rushable.
            return now.isAfter(orderDatePlus30);
        } );
    }
}
```

```
System.out.println("Anvil isPriorityOrder: " +  
    anvil.isPriorityOrder()); // True  
System.out.println("Hammer isPriorityOrder: " +  
    hammer.isPriorityOrder()); // False  
  
    ....  
}  
}
```

Give Intertech a call at **1.800.866.9884** or visit Intertech's website.

