Motivation:

The double negative in the code is very confusing. This makes the code unreadable. It can damage the code which in turn leads to bugs. So, fix them on sight. Here in the example you can see the use of double negative in the codes. In the checkout method there is a if condition which is (! customer.IsNotFlagged) and another one is in the Customer class which is IsNotFlagged. These conditions are really confusing and hard to understand.

**Before:**

public class Order

{

public void Checkout (IEnumerable<Product> products, Customer customer)

{

if (! customer.IsNotFlagged)

{

return;

}

}

}

public class Customer

{

public decimal Balance {get; private set;}

public bool IsNotFlagged

{

get {return Balance < 30m;}

}

}

**Mechanics:**

Here I am going to remove all the negative logic by converting the logic negative to positive and reversing the sense. **(! customer.IsNotFlagged)** this condition will be converted to **(customer.IsFlagged).** And **IsNotFlagged** will be converted to **IsFlagged**

**After:**

public class Order

{

public void Checkout (IEnumerable<Product> products, Customer customer)

{

if (customer.IsFlagged)

{

return;

}

}

}

public class Customer

{

public decimal Balance {get; private set;}

public bool IsFlagged

{

get {return Balance >= 30m;}

}

}