

As mentioned in the lecture, BigDecimal should be used when it comes to monetary calculations. When it comes to monetary calculations, BigDecimal is preferred for its *precision & rounding strategies*. Mostly double should be good for precision concerns, but rounding strategies that BigDecimal class offers out-of-the-box seems to be its big advantage. This is very useful for monetary calculations. Take a look at the following article that explains BigDecimal from monetary standpoint. It explains how one rounding strategy (ROUND_CEILING) is useful when calculating tax while another while charging the cost external to tax (ROUND_HALF_UP). As far as I know, when using double, programmer has to apply such rounding strategies manually and that could be clumsy and errors can creep in. Don't worry if you are unable to follow the examples in the below pointer. You can revisit it after completing couple of more sections in the course.

http://www.opentaps.org/docs/index.php/How_to_Use_Java_BigDecimal:_A_Tutorial

Update: As of May 2022, the above link seems to be not working anymore. In the resources section, a pdf document of the above page has been uploaded and you can check it out.

You can also bookmark the following link as the site discusses various best practices and below link is specific to monetary calculation

<http://www.javapractices.com/topic/TopicAction.do?Id=13>