CSC108H Worksheet: Object-Oriented Programming – Class Day

For each method, implement the body.

```
import event

class Day:
    """A calendar day and its events."""

    def __init__(self, day, month, year):
        """ (Day, int, str, int) -> NoneType

        Initialize a day on the calendar with day, month and year,
        and no events.

        >>> d = Day(5, 'April', 2014)
        >>> d.day
        5
        >>> d.month
        'April'
        >>> d.year
        2014
        >>> d.events
        []
        """
```

```
    def schedule_event(self, new_event):
        """ (Day, Event) -> NoneType

        Schedule new_event on this day, even if it overlaps with
        an existing event. Later we will improve this method.

        >>> d = Day(26, 'March', 2014)
        >>> e = event.Event(11, 12, 'Meeting')
        >>> d.schedule_event(e)
        >>> d.events[0] == e
        True
        """
```

CSC108H Worksheet: Object-Oriented Programming – Class Day

```
    def __str__(self):
        """ (Day) -> str

        Return a string representation of this day.

        >>> d = Day(4, 'April', 2014)
        >>> d.schedule_event(event.Event(13, 14, 'Submit last exercise'))
        >>> d.schedule_event(event.Event(19, 23, 'Celebrate end of classes'))
        >>> print(d)
        4 April 2014:
        - Submit last exercise: from 13 to 14
        - Celebrate end of classes: from 19 to 23
        """
```

```
if __name__ == '__main__':

    # Create day 5 April 2014.

    # Add an event "Sleep in" from 0 to 11 on 5 April 2014.

    # Add an event "Brunch" from 11 to 13 on 5 April 2014.

    # Print the day 5 April 2014, including its events.
```