CSC108H Worksheet: Object-Oriented Programming – Class Day IMPROVED

Below is one solution to the `__init__` method that we wrote for the `Day` class on our previous worksheet. Modify this code to add default parameter values, so that if the date isn't specified by the caller, it is set to January 1, 2014.

```
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
        []
        """

        self.day = day
        self.month = month
        self.year = year
        self.events = []
```

Now, create some events using the default values whenever possible.

```
if __name__ == '__main__':


    # Create New Year's Day 2014.




    # Create your own birthday.




    # Create the first day of classes this term: January 6, 2014




    # Create Canada Day, 2014.
```

CSC108H Worksheet: Object-Oriented Programming – Class Day IMPROVED

Below is the `schedule_event` method that we wrote earlier. We want to improve this method so that when we double-book ourselves by scheduling an event that overlaps with an existing event in the calendar, the method reports this. Change `schedule_event` to return `True` if this new event doesn't overlap with any existing event on this day and `False` if it makes us double-booked.

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

    Schedule new_event on this day.




    >>> d = Day(26, 'March', 2013)
    >>> e = event.Event(11, 12, 'Meeting')
    >>> d.schedule_event(e)

    >>> d.events[0] == e

    """
    self.events.append(new_event)
```

Make one more change to `schedule_event` so that it doesn't even schedule an event if it overlaps with an existing event in our calendar. Remember to change the docstring and the code.

On another sheet of paper, write some test code that creates some overlapping and non-overlapping events and tries to schedule them on the same day.