Below is one solution to the \texttt{\_\_init\_\_} method that we wrote for the \texttt{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.

```python
class Day:
    
    def \_\_init\_\_(self, calendar\_day, calendar\_month, calendar\_year):
        
        Initialize a day on the calendar with calendar\_day, calendar\_month and calendar\_year, and no events.

        >>> d = Day(1, 'August', 2015)
        >>> d.day
        1
        >>> d.month
        'August'
        >>> d.year
        2015
        >>> d.events
        []
        
        self.day = calendar\_day
        self.month = calendar\_month
        self.year = calendar\_year
        self.events = []
```

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

```python
if \_\_name\_\_ == '\_\_main\_\_':

    # Create New Year's Day 2014.

    # Create your own birthday.

    # Create the first day of classes last winter: January 6, 2014

    # Create Canada Day, 2014.
```
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.

```python
def schedule_event(self, new_event):
    """ (Day, Event) ->
    Schedule new_event on this day.
    ""
    self.events.append(new_event)
```

>>> d = Day(3, 'December', 2014)
>>> e = event.Event(17, 23, 'Celebrate end of classes')
>>> d.schedule_event(e)

```python
>>> 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.