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, 2015.

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

    >>> d = Day(1, 'April', 2015)
    >>> d.day
    1
    >>> d.month
    'April'
    >>> d.year
    2015
    >>> d.events
    []
    """

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

```python
if __name__ == '__main__':

    # Create New Year's Day 2015.

    # Create your own birthday.

    # Create the first day of classes this term: January 5, 2015

    # Create Canada Day, 2015. (July 1, 2015)
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(2, 'April', 2015)
>>> e = event.Event(16, 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.