

**Exam 3 Solutions.** Of course, there can be other correct ways to do these problems.

**Problem 1**

```
counter_button['command'] = lambda: increment(counter_button) # in main()
```

```
def increment(button):
    value = button['text']
    new_value = str(1 + int(value))
    button['text'] = new_value
```

**Problem 2**(many other sets of instance variables could work.

Constructor: (many other sets of instance variables and algorithms can also work).

```
def __init__(self, p1, p2):
    self.min_y = min(p1.y, p2.y)
    self.max_y = max(p1.y, p2.y)
    self.min_x = min(p1.x, p2.x)
    self.max_x = max(p1.x, p2.x)

def get_max_x(self):
    return self.max_x

def get_max_y(self):
    return self.max_y

def get_min_x(self):
    return self.min_x

def get_min_y(self):
    return self.min_y

def get_area(self):
    return (self.get_max_x() - self.get_min_x()) * (self.get_max_y() - self.get_min_y())

def contains_point(self, p):
    return self.min_x <= p.x <= self.max_x and self.min_y <= p.y <= self.max_y

def intersects (self, other_rect):
    return self.contains_a_corner_point_of(other_rect) or \
        other_rect.contains_a_corner_point_of(self)

def contains_a_corner_point_of(self, other_rect):
    for x in [other_rect.get_min_x(), other_rect.get_max_x()]:
        for y in [other_rect.get_min_y(), other_rect.get_max_y()]:
            if self.contains_point(Point(x, y)):
                return True
    return False

def intersection(self, other_rect):
    if not self.intersects(other_rect):
        return None
    max_y = min(self.get_max_y(), other_rect.get_max_y())
    max_x = min(self.get_max_x(), other_rect.get_max_x())
    min_y = max(self.get_min_y(), other_rect.get_min_y())
    min_x = max(self.get_min_x(), other_rect.get_min_x())
    return Rectangle(Point(min_x, min_y), Point(max_x, max_y))

def largest_rectangle(rects):
    largest_so_far = rects[0]
    for rect in rects:
        if rect.get_area() > largest_so_far.get_area():
            largest_so_far = rect
    return largest_so_far
```