**1** Explain what the following code fragment does. (1)

```
void draw() {
  if (mouseX < width / 2)
    background(255, 0, 0);
  else
    background(0, 255, 0);
}
```

**2** When you move the mouse quickly, any element relying on `mouseX` and/or `mouseY` for its location seems to "lag" behind by a very short delay. Explain why this occurs. (1)

**3** Under what circumstances would you use the *mouse* for user interaction over the keyboard? When would you choose to use the *keyboard*? (2)

**4** The keyword `keyPressed` evaluates to `true` if a keyboard key is currently pressed and `false` if it is not. Create a program that will draw a random rectangle on the screen every time a key is pressed. (2)

**5** (a) Create a program that will draw a random square on the screen. This square should change colour in response to a mouse click. (4)
   **Note:** The box should change colour only if the mouse is clicked on the box itself, not just anywhere on the screen.

   (b) Explain how you would expand your program to draw 3 squares on the screen that respond to a mouse click. 5? 10?

**6** Create a program that will accomplish each of the following tasks. (6)

   (a) Draw a vertical and horizontal line always intersecting at (`width`, `height`).
   **Hint:** Placing the `noCursor()` command in `setup()` will result in a more pleasant display.
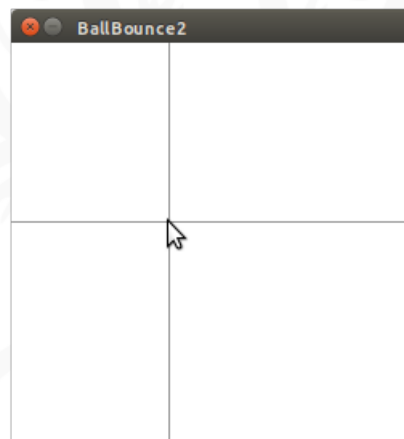


   (b) Adopt the methods implemented in class as part of our `BallBounce` program in order to program a ball that bounces off the newly drawn lines as well as the sides of the window.
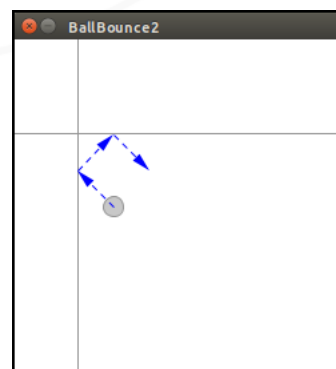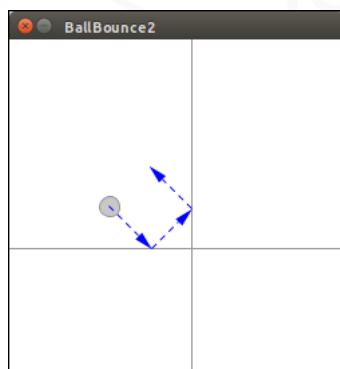


   (c) Modify your program to allow the ball to pass through the drawn lines when the mouse button is held down. Implement some visual clue that indicates the ball will be allowed to pass.