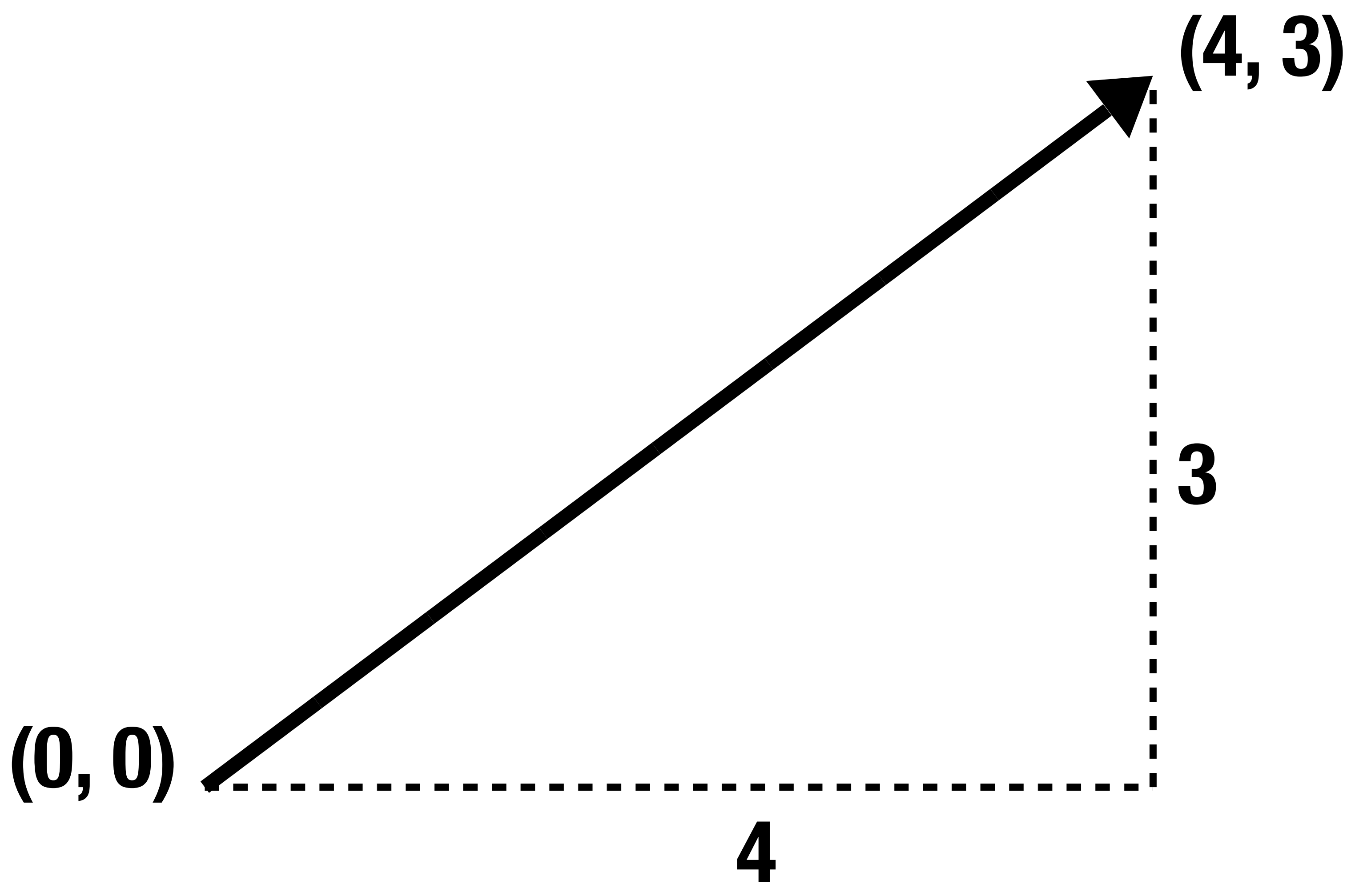
The Vector

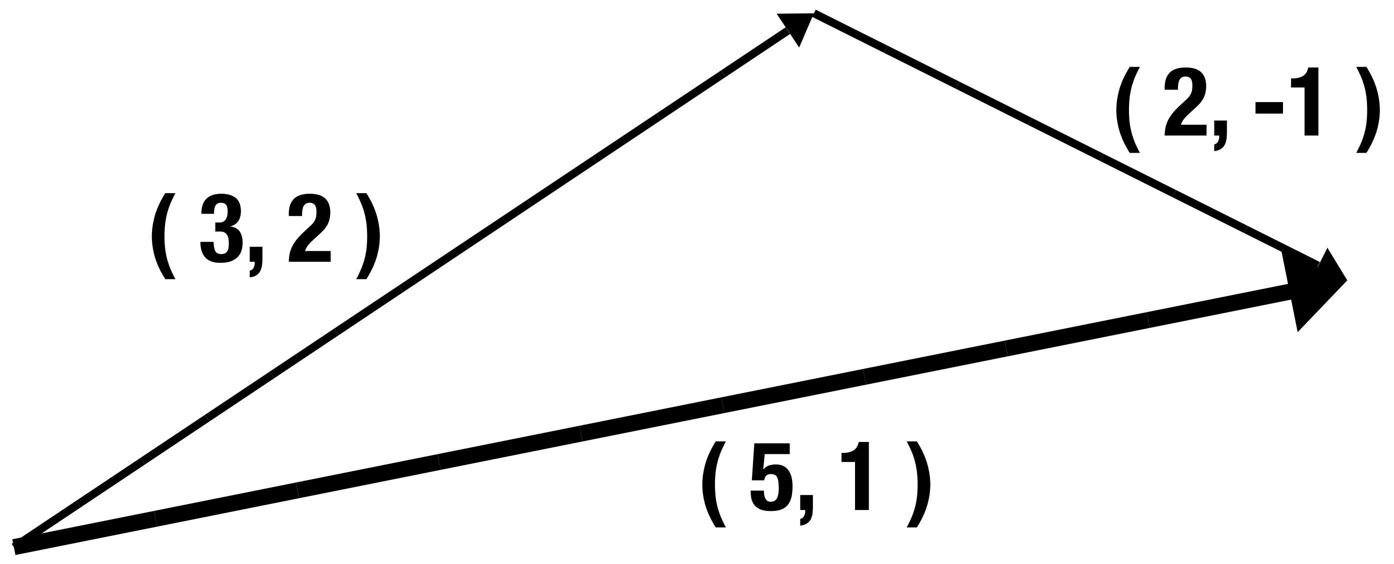
step01_whatdoesavectorlooklike



```
class Vector2f {

  float x = 0;
  float y = 0;
}
```

step02_addingvectors



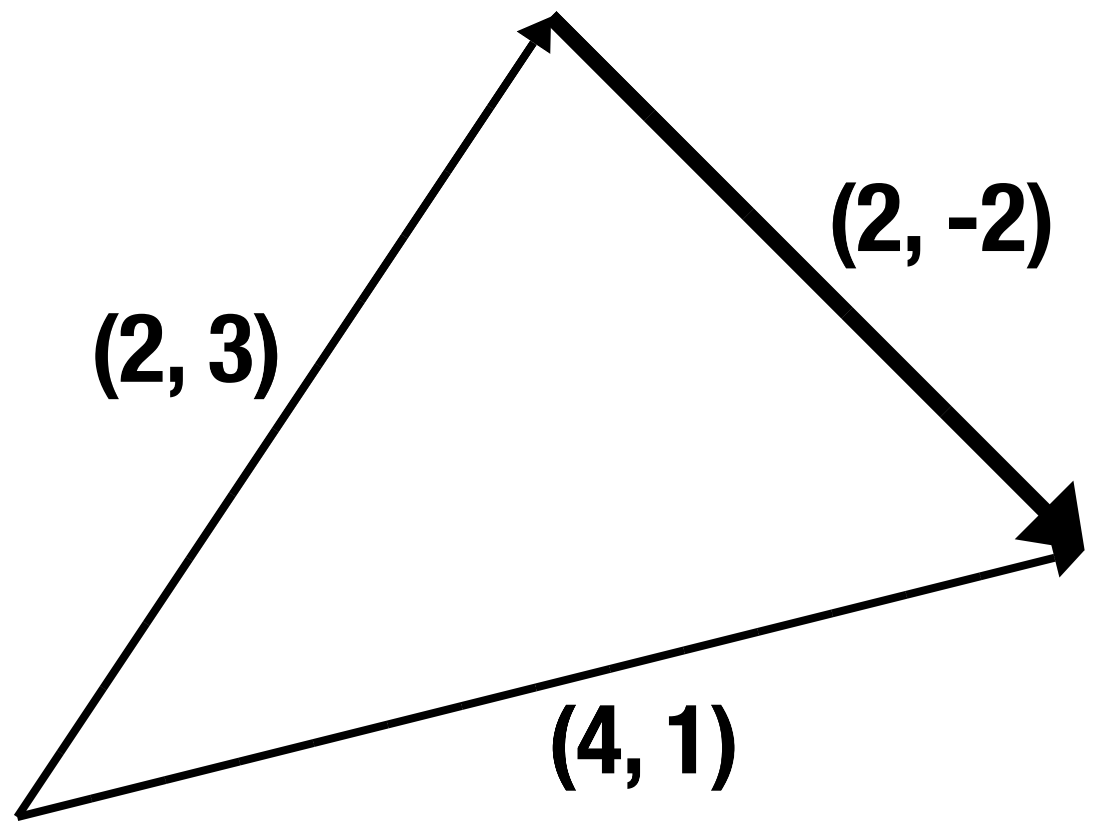
$$(3, 2) + (2, -1) = (5, 1)$$

```
class Vector2f {

  float x = 0;
  float y = 0;

  void add(Vector2f theVector) {
    x += theVector.x;
    y += theVector.y;
  }
}
```

step03_subtractingvectors



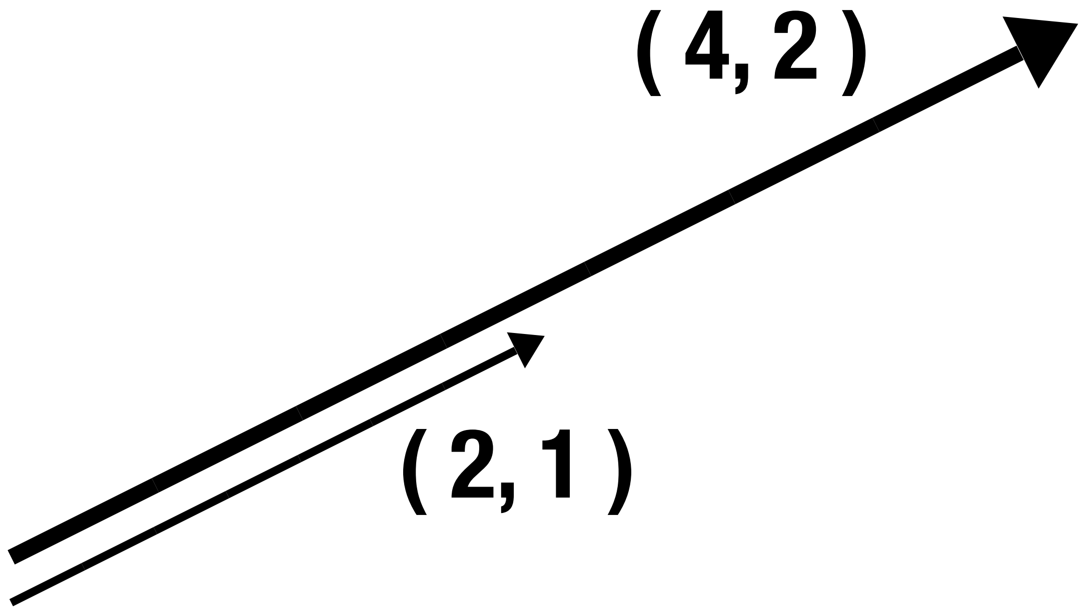
$$(4, 1) - (2, 3) = (2, -2)$$

```
class Vector2f {

  float x = 0;
  float y = 0;

  void sub(Vector2f theVector) {
    x -= theVector.x;
    y -= theVector.y;
  }
}
```

step04_multiplyingvectors



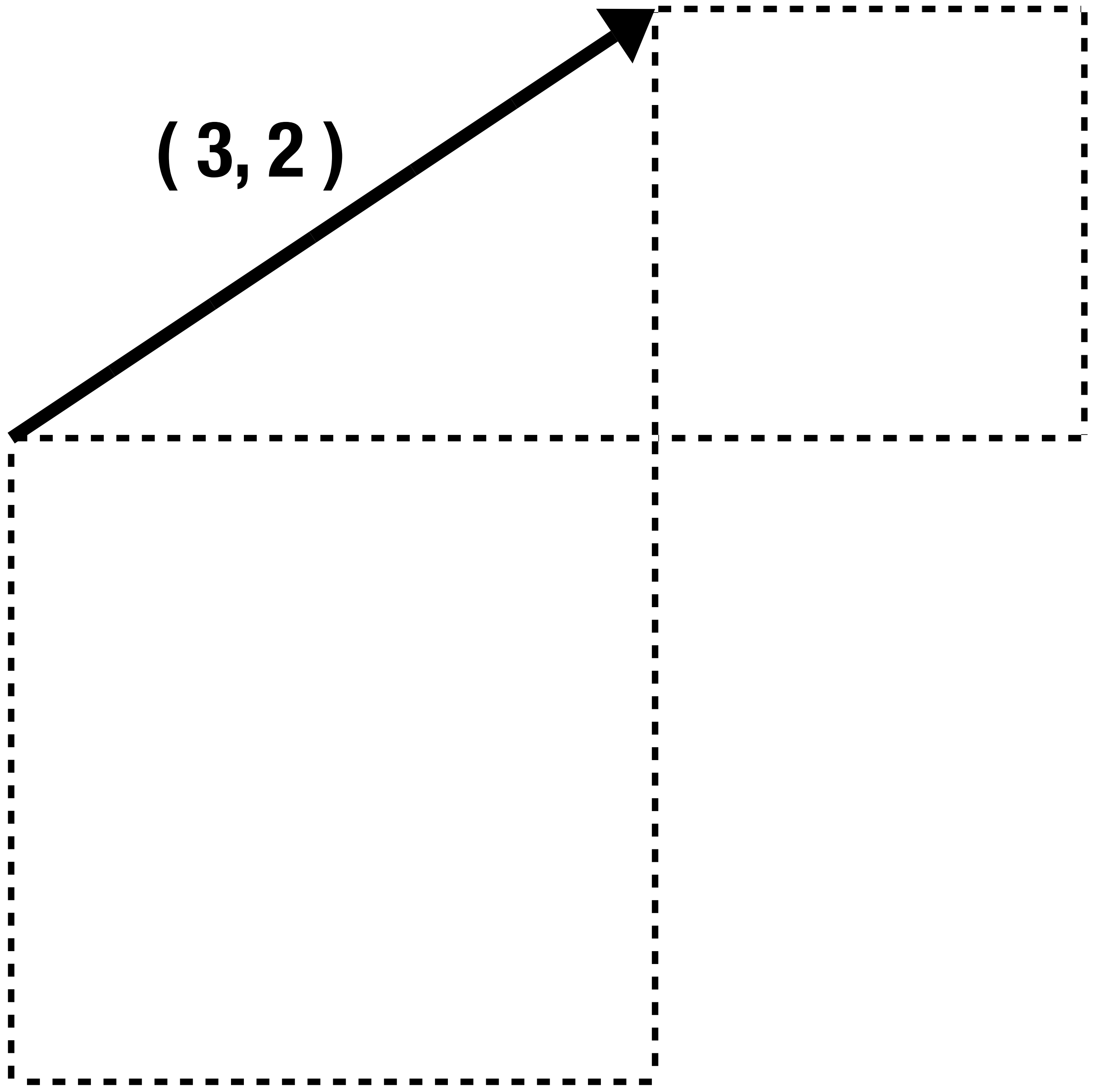
$$(2, 1) * 2 = (4, 2)$$

```
class Vector2f {

  float x = 0;
  float y = 0;

  void multiply(float s) {
    x *= s;
    y *= s;
  }
}
```

step05_lengthofvectors



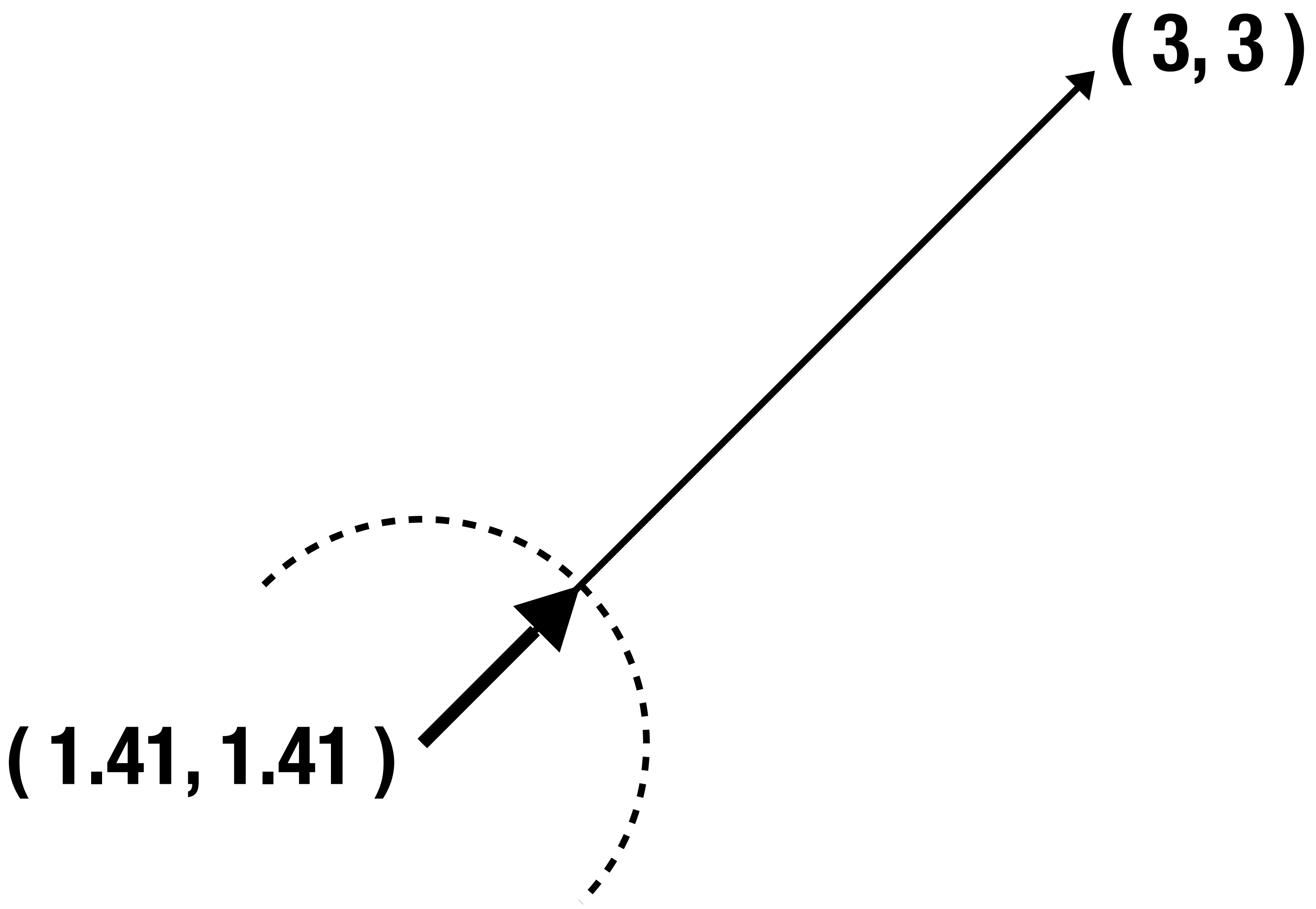
LENGTH = $\sqrt{}$ (3 * 3) + (2 * 2) = $\sqrt{13}$

```
class Vector2f {

  float x = 0;
  float y = 0;

  float length() {
    float myLengthSquard = x*x + y*y;
    float myLength = (float)Math.sqrt(myLengthSquard);
    return myLength;
  }
}
```

step06_normalizingvectors

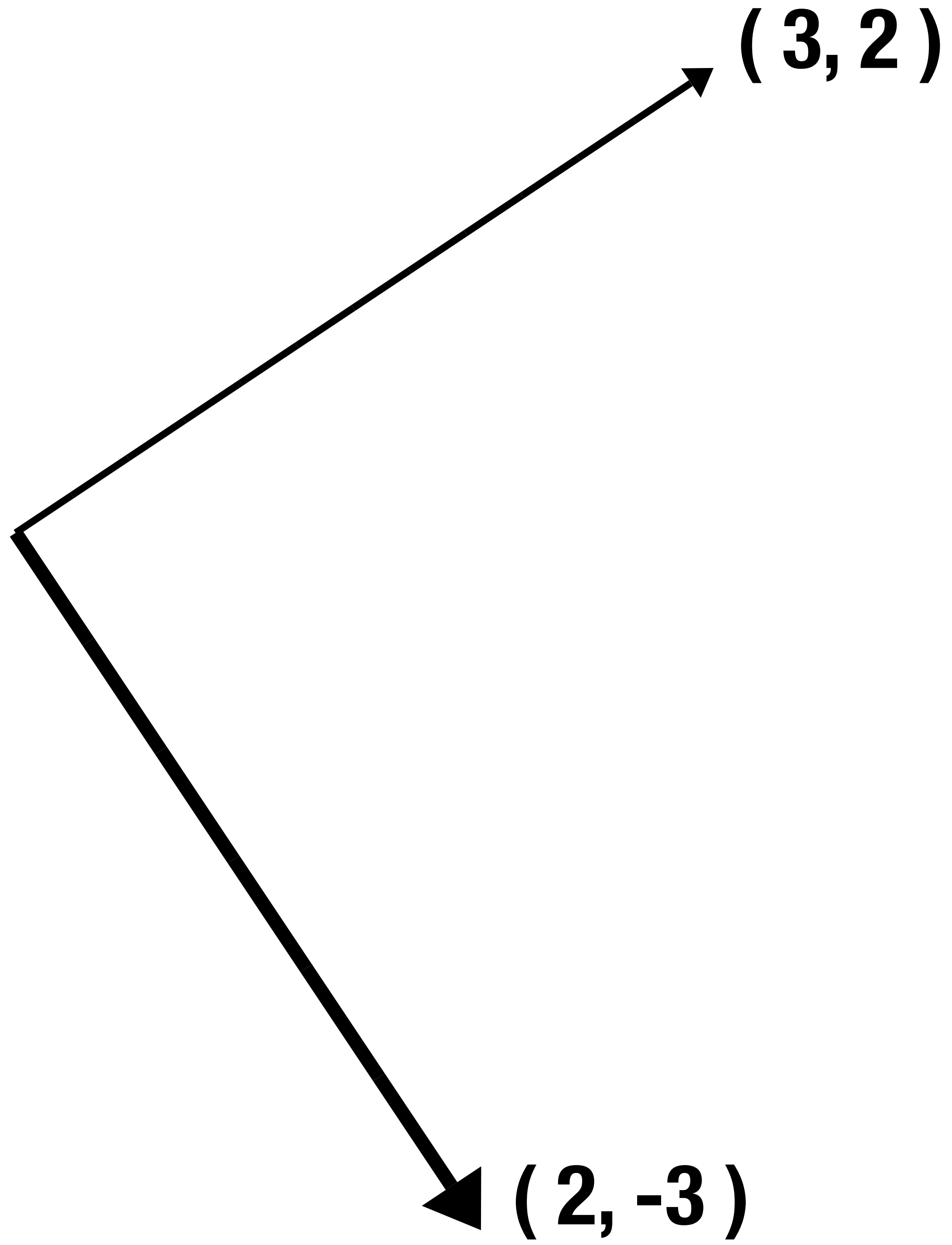


```
class Vector2f {

  float x = 0;
  float y = 0;

  void normalize() {
    float d = length();
    x /= d;
    y /= d;
  }
}
```

step07_crossproduct



```
class Vector2f {

  float x = 0;
  float y = 0;

  void cross(Vector2f a) {
    x = a.y;
    y = -a.x;
  }
}
```

```
class Vector2f { // all together

  float x = 0;
  float y = 0;

  void add(Vector2f theVector) {
    x += theVector.x;
    y += theVector.y;
  }

  void sub(Vector2f theVector) {
    x -= theVector.x;
    y -= theVector.y;
  }

  void multiply(float s) {
    x *= s;
    y *= s;
  }

  float length() {
    float myLengthSquard = x*x + y*y;
    float myLength = (float)Math.sqrt(myLengthSquard);
    return myLength;
  }

  void normalize() {
    float d = length();
    x /= d;
    y /= d;
  }

  void cross(Vector2f a) {
    x = a.y;
    y = -a.x;
  }
}
```