1. Task 2:
   1. The test is failed when it cops with the first case.
   2. The first number is 0 not 1
   3. I change this line “**case** 0: **return** 1;” to “**case** 0: **return** 0;”
2. Task 3:
   1. Error:
      1. Calculation result is incorrect
      2. The error pieces of code is : “**this**.x = y;”
   2. Correction:
      1. Change “**this**.x = y;” to “**this**.x = x;”
   3. My improvement to getDiagonal and getArea

/\*\*

\* Gets the area.

\*

\* **@return** the area

\*/

**public** Double getArea() {

**double** edge1 = Math.*abs*((p2.x - p1.x));

**double** edge2 = Math.*abs*((p2.y - p1.y));

**return** edge1 \* edge2;

}

/\*\*

\* Gets the diagonal.

\*

\* **@return** the diagonal

\*/

**public** Double getDiagonal() {

**double** edge1 = Math.*pow*((p2.x - p1.x), 2);

**double** edge2 = Math.*pow*((p2.y - p1.y), 2);

**return** Math.*sqrt*(edge1 + edge2);

}

1. Task 4: Briefly describe any bugs that I found

The precondition of getBalance, makePurchase, returnChange asks for (balance >= 0)but there was any code to throw exception to this cases. It’s not a normal bug, it’s potential bugs

1. Task 5:
   1. What I learned from this project
      1. JUnit uses assertions to check if the actual result of a function matches the expected value
      2. I used to debug my work with many statements I injected into the code. This new way help me to do less work and could confirm the correctness of my work
      3. I learned many used assertions such as
         1. assertEquals
         2. assertTrue
         3. assertFalse
         4. assertNull
         5. assertSame
      4. It looks more professional when we use assert to replace normal compare way like “ a == b ” or “a == null” or “ a == true”
   2. What I liked and didn’t like about JUnit’s support for unit testing
      1. Like
         1. It’s quite good and easy to use
         2. It help me to reduce the amount of time to debug the project for some silly error such as mistyping …..
      2. Disliked
         1. I think there is no reason to dislike JUnit