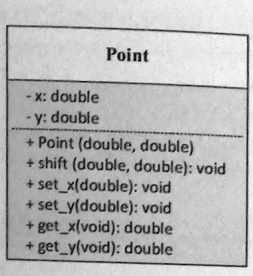
```
6
7       three = one.multiply(two);
8
9       cout << one.getFeet() << endl << two.getFeet() << endl
10          << three.getFeet() << endl;
11
12      return 0;
13  }
```

EXERCISE 3: PROBLEM SOLVING

1. Based on the following class diagram, write a complete C++ program that do the following tasks.



   a. Write the constructor for the class **point** that initialize the values of **x** and **y** with the passed arguments. This constructor also enable default constructor to initialize both values **x** and **y** to 0.

   b. Define the mutator and accessor functions.

   c. Write definition of the member function **`shift()`** that modify values of **x** and **y** by summing up with the passed arguments, **`x_amount`** and **`y_amount`**, respectively.

   d. Write the **`main`** function to create two instance object **`p1`** with parameter setting and **`p2`** with default constructor. Set **`p2`** coordinate with mutator functions and call **`shift()`** function to move the two coordinates to move other quadrant. Make sure the coordinates are printed before and after the move operation.

2. Write a complete C++ program that contains a class called **`Product`**. This class is used for holding data about products sold by a grocery store. The class should have the following **`private`** member variables (attributes):

Attribute	Description
`code`	is of type `int` that holds the product's code number.
`price`	is of type `double` that holds the price of the product.

The class should also have the following **`public`** member variables (methods):

Attribute	Description
Default constructor	Sets all the attributes to `0`.
Overloaded Constructor	Sets all the attributes from arguments.
Destructor	Does nothing.
`setCode`	is a mutator function that sets the **`code`** attribute from an **`int`** argument.
`setPrice`	is a mutator function that sets the **`price`** attribute from a **`double`** argument.
`getCode`	is an accessor function that returns the value of **`code`** attribute.
`getPrice`	is an accessor function that returns the value of **`price`** attribute.

In the **`main`** function of the program:

   a. Declare an array to hold a list of up to 100 objects of class **`Product`**.

   b. Ask the user to enter the number of products to key in.

   c. Set the attributes of each object in the array using the mutator functions. Data to assign are firstly read from the keyboard. Your program is expected to use a loop for doing this.

   d. Add another Product object to the array. The objects's code and price attributes are set to 100 and 1.00 respectively. Your program is expected to use the overloaded constructor for creating this object.

   e. Finally, print the product information in a table as shown in the Figure 3.2. Your program should use another loop for doing this part. Your program should also calculate the average price of the products and print it at the end of the table.

The **bold text** in the following figure represent user inputs from the keyboard.

```
How many products to enter? 2

Enter the information of product #1
Product Code: 101
Price: 4.50

Enter the information of product #2
Product Code: 933
Price: 8.90

No      Product Code    Price

1       101             4.50
2       933             8.90

Average Price:  RM 6.70
```

Figure 3.2