```java
// Fig. 3.1: Account.java
// Account class that contains a name instance variable
// and methods to set and get its value.

public class Account
{
   private String name; // instance variable

   // method to set the name in the object
   public void setName(String name)
   {
      this.name = name; // store the name
   }

   // method to retrieve the name from the object
   public String getName()
   {
      return name; // return value of name to caller
   }
} // end class Account
```

Fig. 3.1 | `Account` class that contains a `name` instance variable and methods to *set* and *get* its value.

Example 1: Client class to Account class

```java
// Fig. 3.2: AccountTest.java
// Creating and manipulating an Account object.
import java.util.Scanner;

public class AccountTest
{
   public static void main(String[] args)
   {
      // create a Scanner object to obtain input from the command window
      Scanner input = new Scanner(System.in);

      // create an Account object and assign it to myAccount
      Account myAccount = new Account();

      // display initial value of name (null)
      System.out.printf("Initial name is: %s%n%n", myAccount.getName());

      // prompt for and read name
      System.out.println("Please enter the name:");
      String theName = input.nextLine(); // read a line of text
      myAccount.setName(theName); // put theName in myAccount
      System.out.println(); // outputs a blank line

```

Fig. 3.2 | Creating and manipulating an `Account` object. (Part 1 of 2.)

Example 1: Client class to Account class Cont'd

```
24         // display the name stored in object myAccount
25         System.out.printf("Name in object myAccount is:%n%s%n",
26            myAccount.getName());
27      }
28   } // end class AccountTest
```

```
Initial name is: null

Please enter the name:
Jane Green

Name in object myAccount is:
Jane Green
```

Fig. 3.2 | Creating and manipulating an Account object. (Part 2 of 2.)

Note:

- ✓ A Single java application can contain one or more packages.
- ✓ Each package contains can contain one or more classes
- ✓ The access modifier of a class can be public or default (not public)
- ✓ If a class has no package name, the class belongs to default package
- ✓ **The name of the file of the source code is affected by the class modifier**
- ✓ **The communication of a client class and a server class is affected by class modifier and package names**.
- ✓ The following tasks helps you to understand the above facts by compiling and running on the command window.

Instructions of Task 1:

1. Create a folder (directory) on **D** drive with name **Source1**.
2. Write the Fig.3.1 and Fig.3.2 on notepad and save them in **Source1** directory by file names: **Account.java** and **AccountTest.java**
3. Compile the source codes from command line and check that corresponding class files **Account.class** and **AccountTest.class are** created in the **Source1** directory.
4. Run the application from command line and attach the screen shot

Instructions of Task 2 (about class name and .java file name)

1. Create a folder (directory) on D drive with name **Source2**.
2. Write Fig.3.1 on notepad and save it as **ACCOUNT.java** in Source2 Directory
3. Write Fig.3.2 on notepad and save it as **AccountTest.java in** Source2 directory
4. Compile the source codes. Is there a completion error? If you say yes, what is the cause of the error and how do you fix the error? Fix the error and attach the screenshot of the program

Instructions of Task 3 (about class name and .class file name)

1. Create a folder (directory) on D drive with name **Source3.**
2. **Modify** Fig.3.1 on notepad by **deleting** the "public" modifier of the **class** and Save the file **Account.java** in Source3 directory
3. Write Fig.3.2 on notepad and save it as **AccountTest.java Source3 directory**
4. Compile the source codes from command line. Is there compilation error? If you say no, why there is no compilation error?
5. If there is no compile time error, run the program from command line and attach the screen shot

Instructions of Task 4(about setting ClassPath)

1. Create a folder (directory) on D drive with name **Source4 and Source_4**.
2. Write Fig.3.1 and Fig.3.2 on notepad and save them in Source4 by file names: **Account.java** and **AccountTest.java**
3. Compile the source codes from command line.
4. **Change** the current directory of the command prompt to **Source_4 directory** that does not contain the class files (.class) generated by step 3.
5. Run the application from the command line under Source_4 directory. **Did you get runtime error**? What is the cause of the error? Fix the error and attach the screenshot of the program

Instructions of Task 5:(communication between a class in one default package and a public class in another non-default package)

1. Create a folder (directory) on D drive with name **Source5**
2. Write Fig.3.1. on notepad without any modification and save in Source5 Directory
3. Re-write Fig.3.2 by adding **package** name **called "AccountTestPachage" and save in Source5 directory**
4. Compile the source codes. Did you get compile error? What is the cause of the error? Fix the error and attach the screenshot of the program

Instructions of Task 6: (communication between a class in a non-defualt package and a public class in another non-default package)

1. Create a folder (directory) on D drive with name **Source6**
2. Rewrite Fig.3.1. on notepad by adding **package** name **called "AccountPackage" and save it in Source6 directory**

3. Re-write Fig.3.2 by adding **package** name **called "AccountTestPachage" and save it in Source6 directory**
4. Compile the source codes. Did you get compile error? What is the cause of the error? How did you correct the error? Fix the error and attach the screenshot of the program

Instructions of Task 7 :(communication between a class in a non-defualt package and a defualt class in another non-default package)

1. Create a folder (directory) on D drive with name **Source7**
2. Rewrite Fig.3.1. on notepad
   - ✓ by adding **package** name **called "AccountPackage"**
   - ✓ **deleting the public modifier of the class and save in Source7 directory**
3. Re-write Fig.3.2 by adding **package** name **called "AccountTestPachage" in source7 directory**
4. Compile the source codes. Did you get compile error? What is the cause of the error? How did you correct the error? Fix the error and attach the screenshot of the program