LAB 1 – Introduction to Object Oriented Programming

Objectives:

At the end of this lab, the students are able to:

i. Differentiate between structural and object-oriented programming.
ii. Create a simple program using object oriented paradigm.
iii. Read and interpret the UML diagram and convert the notation into object-oriented programming.

1.1 Activity 1

1.1.1 Objective

To write a simple program using a structural approach.

1.1.2 Problem Description

You have been assigned to write a program to perform an arithmetic addition and subtraction for two (2) integer numbers key-in by the user. The program should only accept the number key-in by the user as a positive number. Finally, display the result for addition and subtraction. You should write your program using an object- oriented approach.

[Estimated Time: 15 minutes]

1.1.3 Solution Activity 1

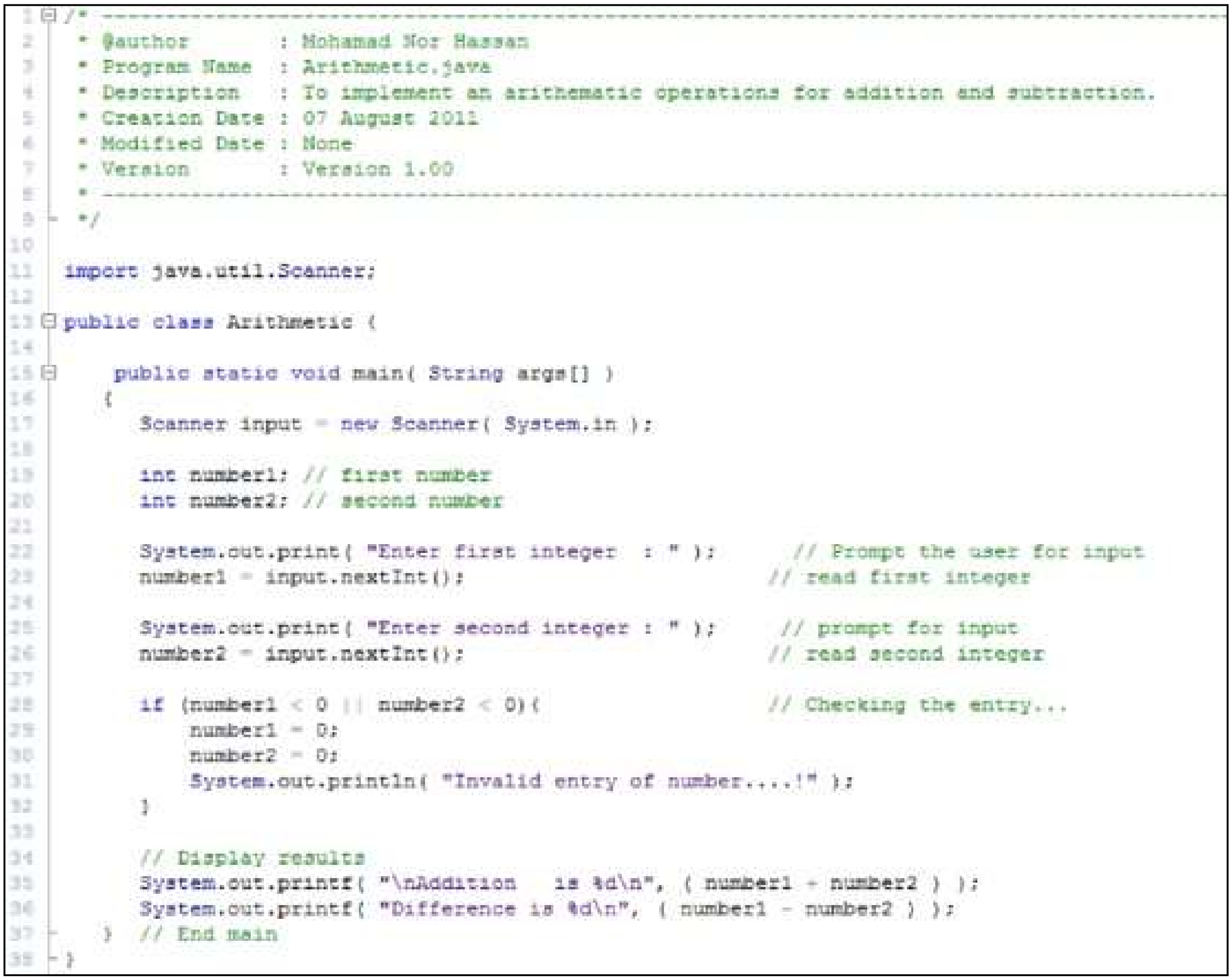
Step 1: Create Working folder called CSF3043 and subfolder called Lab – Chapter 1.

Step 2: Go to CSF3043 –>Lab –Chapter 1's folder and create Sub-working folder called Activity 1.

Step 3: Open IDE NetBeans or JCreator.

Step 4: Create new file/class called Arithmetic.java.

Step 5: Complete the coding as below.



```java
/* -----------------------------------------------------------------------------------------
 * @author       : Mohamad Nor Hassan
 * Program Name  : Arithmetic.java
 * Description   : To implement an arithematic operations for addition and subtraction.
 * Creation Date : 07 August 2011
 * Modified Date : None
 * Version       : Version 1.00
 * -----------------------------------------------------------------------------------------
 */

import java.util.Scanner;

public class Arithmetic {

    public static void main( String args[] )
   {
      Scanner input = new Scanner( System.in );

      int number1; // first number
      int number2; // second number

      System.out.print( "Enter first integer  : " );      // Prompt the user for input
      number1 = input.nextInt();                        // read first integer

      System.out.print( "Enter second integer : " );     // prompt for input
      number2 = input.nextInt();                        // read second integer

      if (number1 < 0 || number2 < 0){                  // Checking the entry...
          number1 = 0;
          number2 = 0;
          System.out.println( "Invalid entry of number....!" );
      }

      // Display results
      System.out.printf( "\nAddition   is %d\n", ( number1 + number2 ) );
      System.out.printf( "Difference is %d\n", ( number1 - number2 ) );
   } // End main
}
```

Step 6: Save the Arithmetic.java file.

Step 7: Compile the program.

Step 8: Run and evaluate the output.

1.2 Activity 2

1.2.1 Objective

To write a simple program using an object-oriented approach.

1.2.2 Problem Description

You have been assigned to write a program to perform an arithmetic addition and subtraction for two (2) integer numbers key-in by the user. The program should only accept the number key-in by the user as a positive number. Finally, display the result for addition and subtraction. You should write your program using an object-oriented approach.

[Estimated Time: 25 minutes]

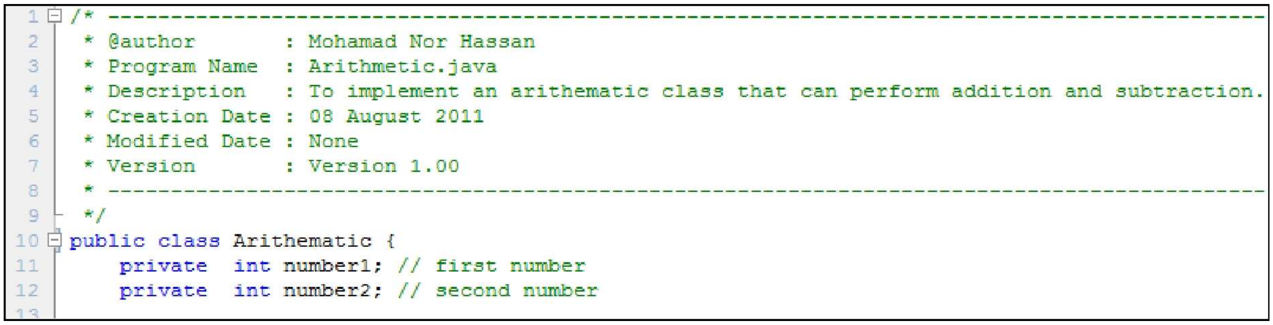
1.2.3 Solution Activity 2

Step 1: Go to CSF3043 –> Lab – Chapter 1's folder and create Sub-working folder called Activity 2.

Step 2: Open IDE Netbeans or JCreator.

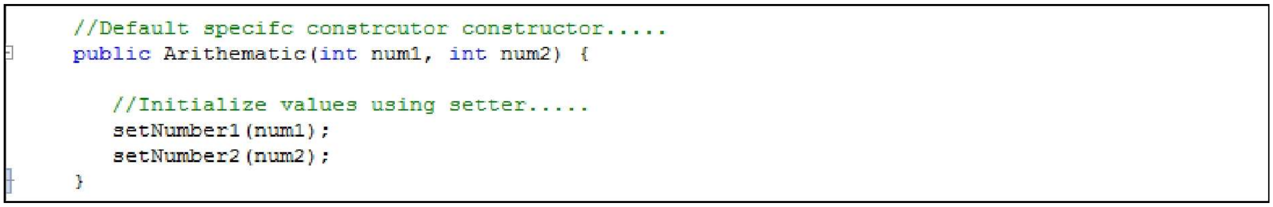
Step 3: Create new file/class called Arithmetic.java.

Step 4: Define the class and instance variables as below:



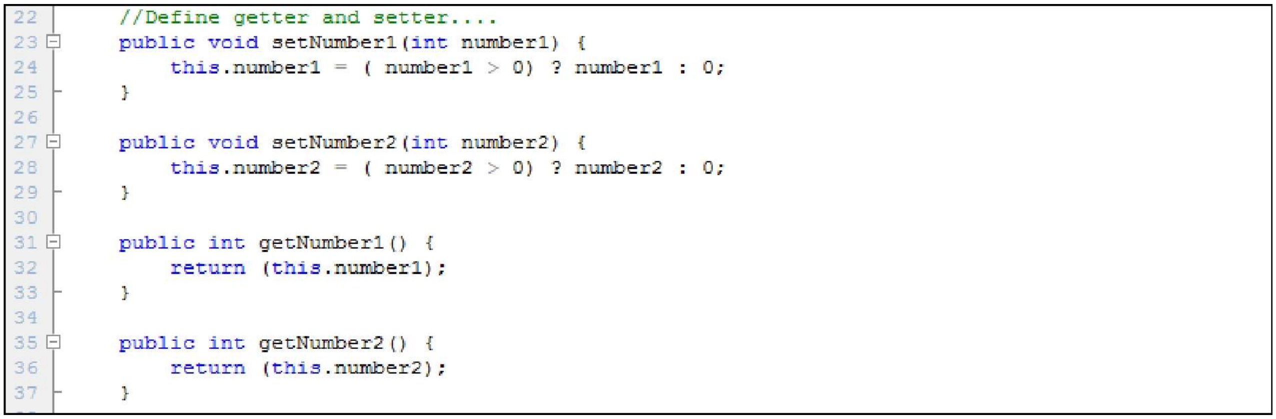
```java
/* ---------------------------------------------------------------------------------------------
 * @author        : Mohamad Nor Hassan
 * Program Name   : Arithmetic.java
 * Description    : To implement an arithematic class that can perform addition and subtraction.
 * Creation Date  : 08 August 2011
 * Modified Date  : None
 * Version        : Version 1.00
 * ---------------------------------------------------------------------------------------------
 */
public class Arithematic {
    private  int number1; // first number
    private  int number2; // second number
```

Step 5: Define the Arithmetic's constructor.



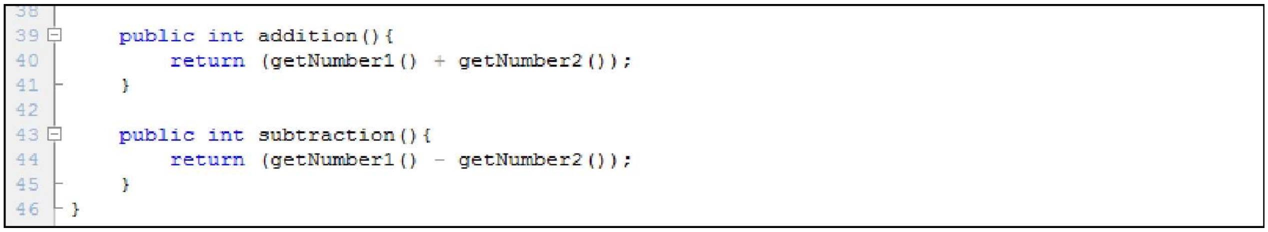
```java
//Default specifc constrcutor constructor.....
public Arithematic(int num1, int num2) {

   //Initialize values using setter.....
   setNumber1(num1);
   setNumber2(num2);
}
```

Step 6: Define the getter and setter for each of instance variable.



```
    //Define getter and setter....
    public void setNumber1(int number1) {
        this.number1 = ( number1 > 0) ? number1 : 0;
    }

    public void setNumber2(int number2) {
        this.number2 = ( number2 > 0) ? number2 : 0;
    }

    public int getNumber1() {
        return (this.number1);
    }

    public int getNumber2() {
        return (this.number2);
    }
```

Step 7: Define the method to perform addition and subtraction.



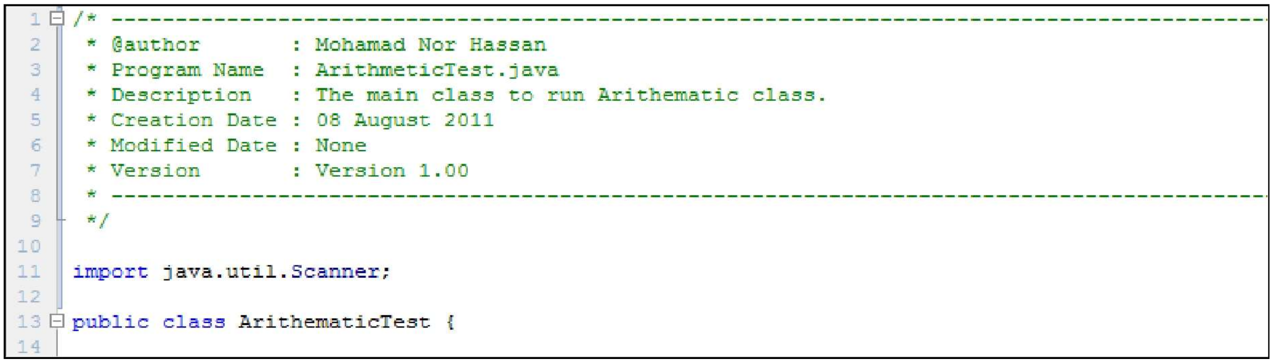
```
    public int addition(){
        return (getNumber1() + getNumber2());
    }

    public int subtraction(){
        return (getNumber1() - getNumber2());
    }
}
```

Step 8: Save the file in sub-working folder called Activity 2.

Step 9: Compile the program.

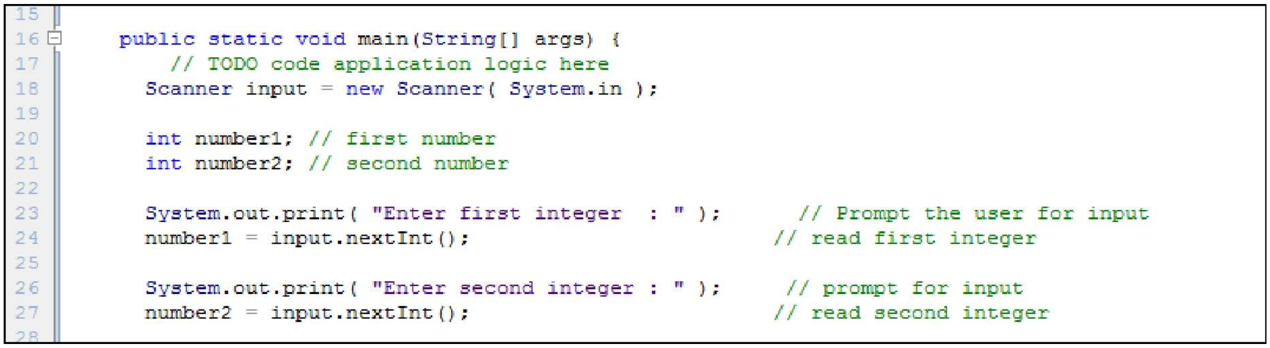
Step 10: Create new file/class called ArithmeticTest.java

Step 11: Used java utility for Scanner and define the ArithmeticTest class as below:



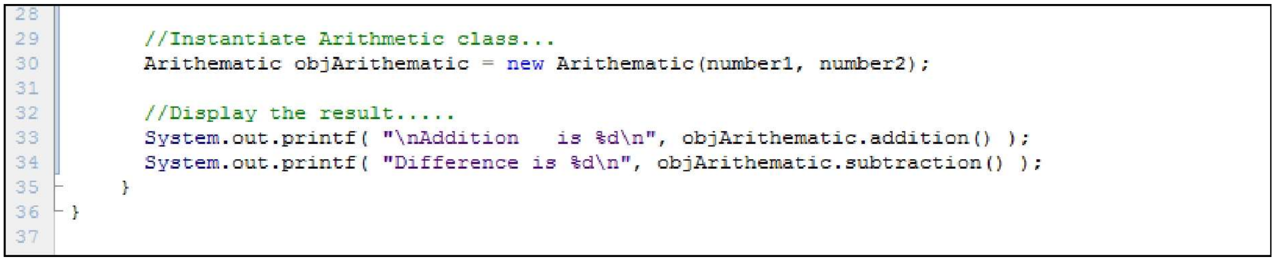
```
/* ----------------------------------------------------------------------------------------
 * @author        : Mohamad Nor Hassan
 * Program Name  : ArithmeticTest.java
 * Description   : The main class to run Arithematic class.
 * Creation Date : 08 August 2011
 * Modified Date : None
 * Version       : Version 1.00
 * ----------------------------------------------------------------------------------------
 */

import java.util.Scanner;

public class ArithematicTest {

```

Step 12: Define main program, instantiate the Scanner class, variables to store data key-in by the user.



```
    public static void main(String[] args) {
        // TODO code application logic here
      Scanner input = new Scanner( System.in );

      int number1; // first number
      int number2; // second number

      System.out.print( "Enter first integer  : " );      // Prompt the user for input
      number1 = input.nextInt();                         // read first integer

      System.out.print( "Enter second integer : " );     // prompt for input
      number2 = input.nextInt();                         // read second integer
```

Step 13: Instantiate the Arithmetic class and display the value for addition and subtraction respectively.



```
      //Instantiate Arithmetic class...
      Arithematic objArithematic = new Arithematic(number1, number2);

      //Display the result.....
      System.out.printf( "\nAddition   is %d\n", objArithematic.addition() );
      System.out.printf( "Difference is %d\n", objArithematic.subtraction() );
    }
}
```

Step 14: Save the file in sub-working folder Lab – Chapter 1 ->Activity 2.

Step 15: Compile the program.

Step 16: Run and evaluate the output.

1.3 Activity 3

1.3.1 Objective

Creating the constructor and simple methods.

1.3.2 Problem Description

Calculate the perimeter and area of the rectangle given that the length of the rectangle is 4 cm and the width is 2 cm. Write your solution by initializing the length and width via constructor. Subsequently, introduce the method for *area()* and *perimeter()*. Finally, in the main program, display the value for area and perimeter for the rectangle. Compile your program and evaluate the output.

[Estimated Time: 40 minutes]

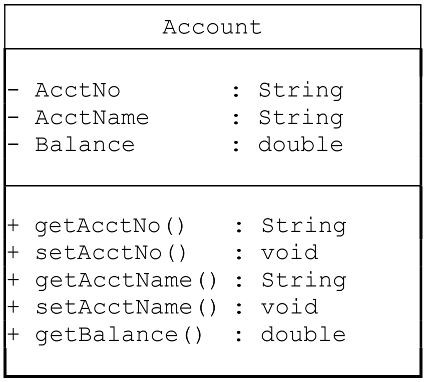
1.4 Activity 4

1.4.1 Objective

Understand the simple UML diagram and convert the UML into object-oriented programming.

1.4.2 Problem Description

Figure 1 show the UML diagram for Account's class. Based on UML diagram, display the account number, account name and balance for account number = "1001". Write your solution using object-oriented approach.



```
Account

- AcctNo          : String
- AcctName        : String
- Balance         : double

+ getAcctNo()     : String
+ setAcctNo()     : void
+ getAcctName()   : String
+ setAcctName()   : void
+ getBalance()    : double
```

Figure 1 – Class Account

[Estimated Time: 40 minutes]

Algorithms:

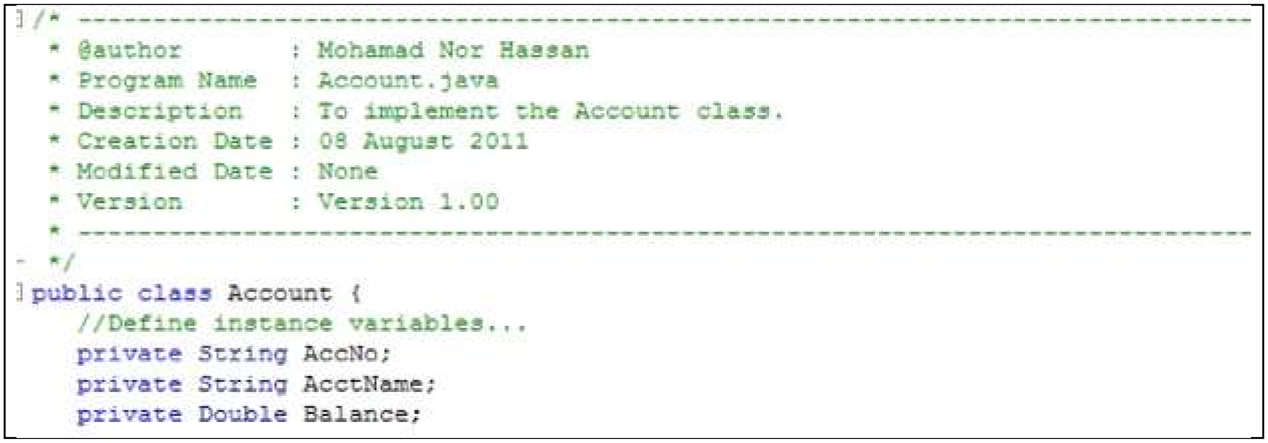
Step 1 – Declare a class Account and attributes.
Step 2 – Create constructor
Step 3 – Initialize the values of account number, name and balance via constructor
Step 4 – Define getter and setter for each attributes
Step 5 - Declare a class AccountTest.
Step 6 – Instantiate Account class and display the account number, name and balance.

1.4.3 Solution Activity 4

Step 1: Go to CSF3043 – > Lab – Chapter 1 folder and create Sub-working folder called Activity 4.

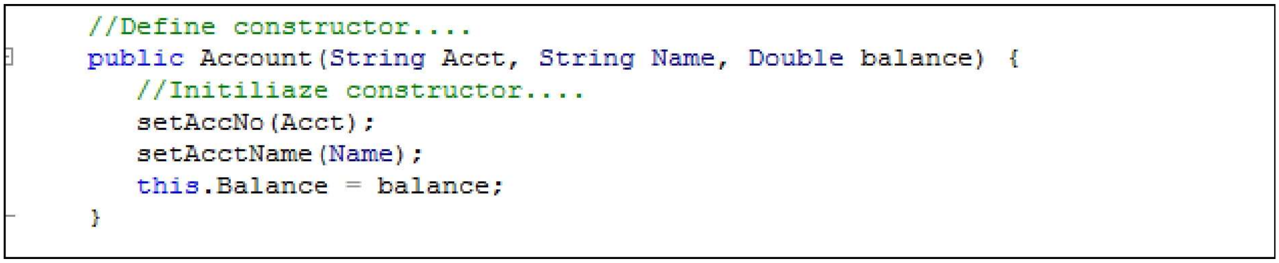
Step 2: Open IDE NetBeans or JCreator.

Step 3: Create new file/class called Account.java.
Step 4: Define the class and instance variables as below:



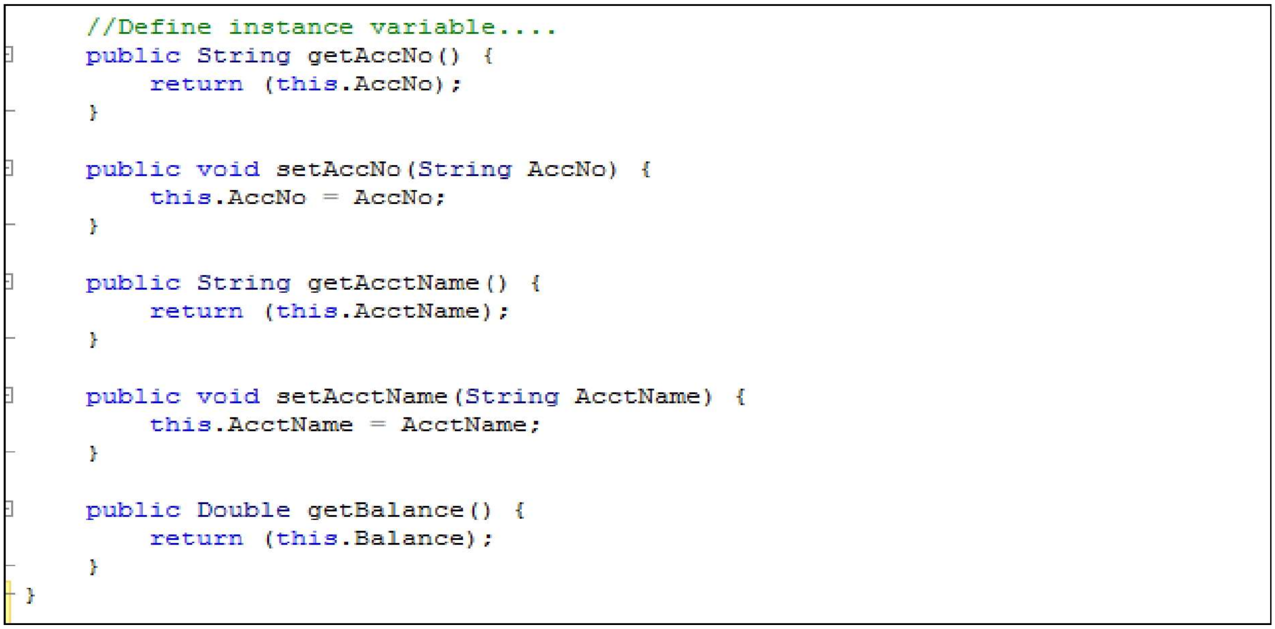
```
/* ---------------------------------------------------------------------------
 * @author         : Mohamad Nor Hassan
 * Program Name    : Account.java
 * Description     : To implement the Account class.
 * Creation Date : 08 August 2011
 * Modified Date : None
 * Version         : Version 1.00
 * ---------------------------------------------------------------------------
 */
public class Account {
   //Define instance variables...
   private String AccNo;
   private String AcctName;
   private Double Balance;
```

Step 5: Define and initialize the constructor for class Account.



```
    //Define constructor....
    public Account(String Acct, String Name, Double balance) {
       //Initiliaze constructor....
       setAccNo(Acct);
       setAcctName(Name);
       this.Balance = balance;
    }
```

Step 6: Define the getter and setter based on the specification written in UML diagram.



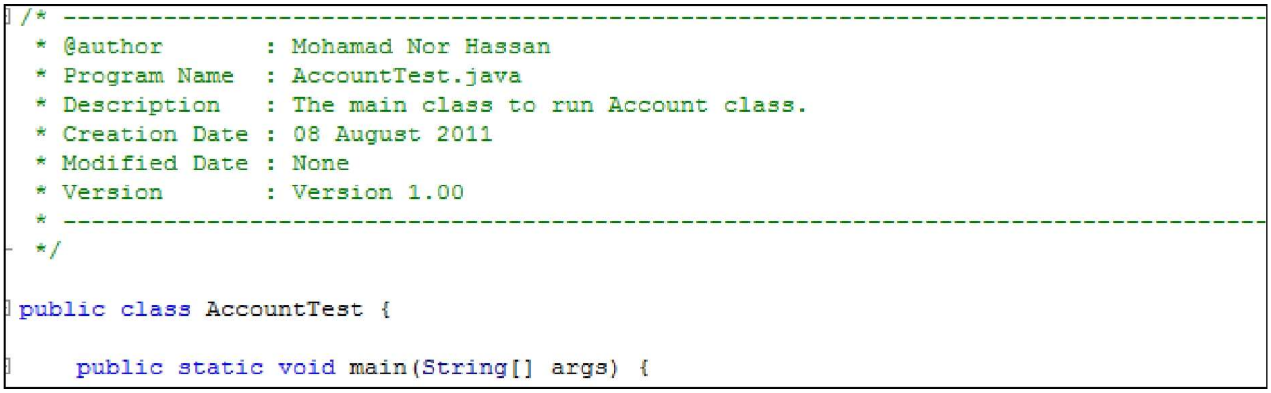
```
    //Define instance variable....
    public String getAccNo() {
        return (this.AccNo);
    }

    public void setAccNo(String AccNo) {
        this.AccNo = AccNo;
    }

    public String getAcctName() {
        return (this.AcctName);
    }

    public void setAcctName(String AcctName) {
        this.AcctName = AcctName;
    }

    public Double getBalance() {
        return (this.Balance);
    }
}
```

Step 7: Save the file in sub-working folder called Activity 4.

Step 9: Compile the program.

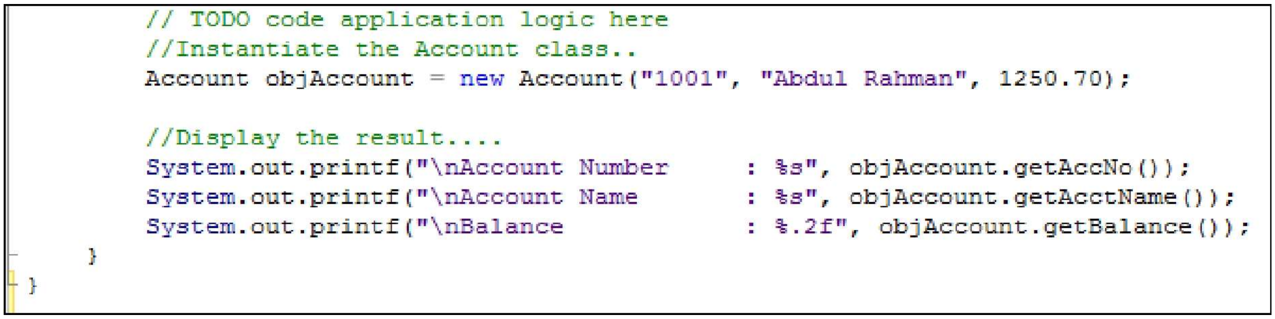
Step 10: Create new file/class called AccountTest.java

Step 11: Define the AccountTest class as below:



```
/* -----------------------------------------------------------------------------------
 * @author        : Mohamad Nor Hassan
 * Program Name   : AccountTest.java
 * Description    : The main class to run Account class.
 * Creation Date  : 08 August 2011
 * Modified Date  : None
 * Version        : Version 1.00
 * -----------------------------------------------------------------------------------
 */

public class AccountTest {

    public static void main(String[] args) {
```

Step 12: Instantiate the Account class and display the account number, account name and balance for account number = “1001”.



```
        // TODO code application logic here
        //Instantiate the Account class..
        Account objAccount = new Account("1001", "Abdul Rahman", 1250.70);

        //Display the result....
        System.out.printf("\nAccount Number     : %s", objAccount.getAccNo());
        System.out.printf("\nAccount Name       : %s", objAccount.getAcctName());
        System.out.printf("\nBalance            : %.2f", objAccount.getBalance());
    }
}
```

Step 13: Save the file in sub-working folder Lab – Chapter 1 ->Activity 4.

Step 14: Compile the program.

Step 15: Run and evaluate the output

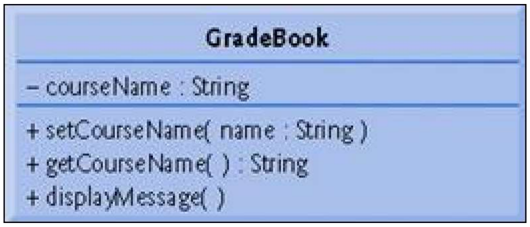
Lab Exercises

Objective

To define the class, constructor, instance variables and methods used to display the output key-in by the user via the main program.

Problem Description

1. Given the UML class diagram below is designed for a simple program that displays the Gradebook course name as key-in by the user.

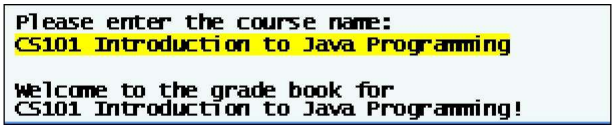


2. The specifications of the program are given below;

   i. Program will prompt *“Please enter the course name”*.

   ii. User will key-in the course name and press ENTER button.

   iii. The program will display the output at window console as below;



```
Please enter the course name:
CS101 Introduction to Java Programming

Welcome to the grade book for
CS101 Introduction to Java Programming!
```

[Estimated Time: 50 minutes]