LAB 2 – Introduction to Class and Object

Objectives

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

i. Define the class for define business object in real life scenario.
ii. Instantiate the object for specific class based on specification define via constructor.
iii. Initialize and using getter and setter to assign and retrieve the value from the main class or different class.
iv. Using primitive data type in class.

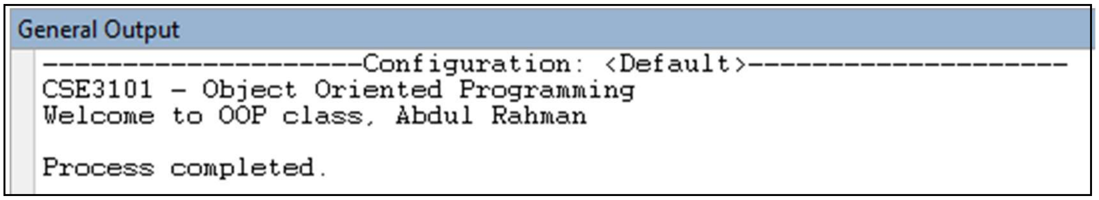
2.1 Activity 1

2.1.1 Objective

Writing a class with simple attributes, constructor and methods.

2.1.2 Problem Description

Write a program to display the message “Welcome to OOP class, *person_name*”. The name of person must be assigned through the constructor. In addition, the program should prompt message “CSF3043 – Object Oriented Programming”, when you execute the default constructor. Run your program using default constructor and your pre-defined constructor. Evaluate the output. You should produce the sample output as below:



General Output

```
--------------------Configuration: <Default>--------------------
CSE3101 - Object Oriented Programming
Welcome to OOP class, Abdul Rahman

Process completed.
```

[Estimated Time: 30 minutes]

2.1.3 Solution Activity 1

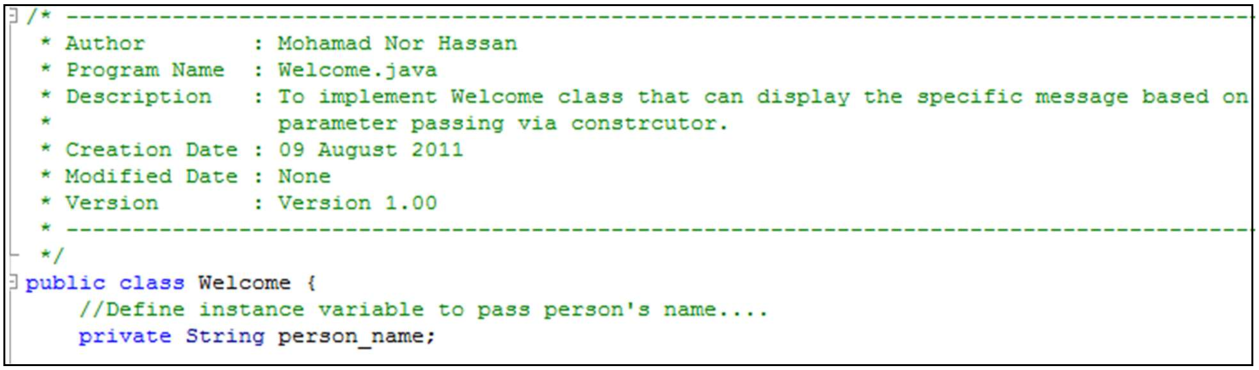
Step 1: Go to CSF3043 folder and create sub-folder called Lab – Chapter 2.

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

Step 3: Open IDE NetBeans or JCreator.

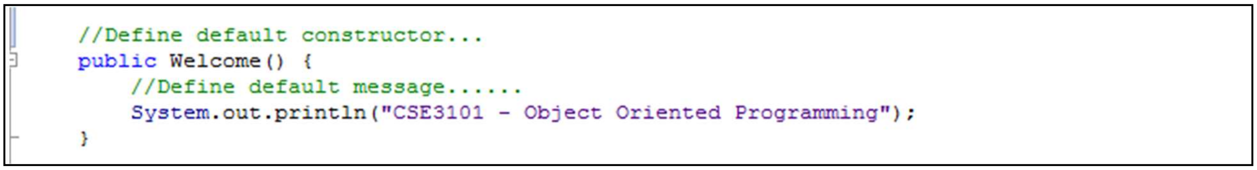
Step 4: Create new file/class called Welcome.java.

Step 5: Define the class and instance variables as below:



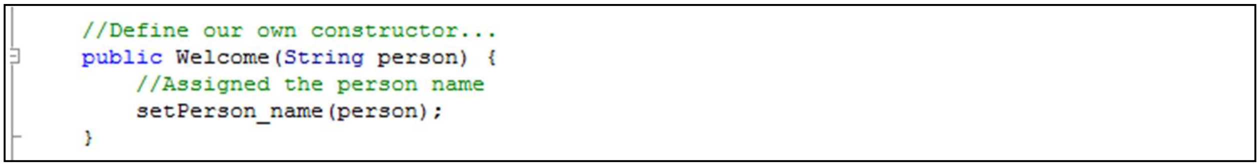
```
/* --------------------------------------------------------------------------------
 * Author        : Mohamad Nor Hassan
 * Program Name  : Welcome.java
 * Description   : To implement Welcome class that can display the specific message based on
 *                 parameter passing via constrcutor.
 * Creation Date : 09 August 2011
 * Modified Date : None
 * Version       : Version 1.00
 * --------------------------------------------------------------------------------
 */
public class Welcome {
    //Define instance variable to pass person's name....
    private String person_name;
```

Step 6: Define the default constructor for Welcome class.



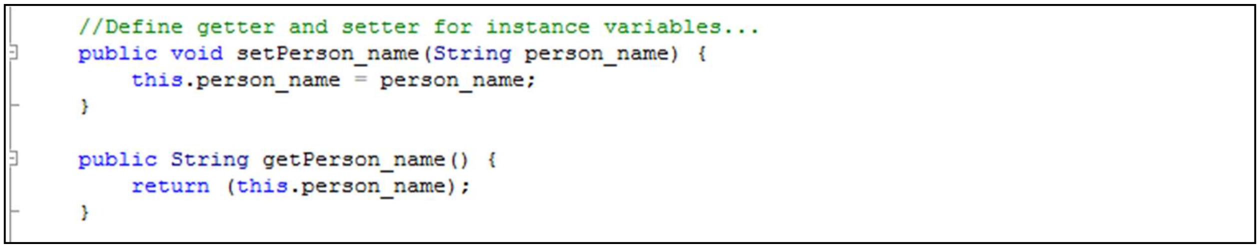
```
    //Define default constructor...
    public Welcome() {
        //Define default message......
        System.out.println("CSE3101 - Object Oriented Programming");
    }
```

Step 7: Define 2nd constructor for Welcome class.



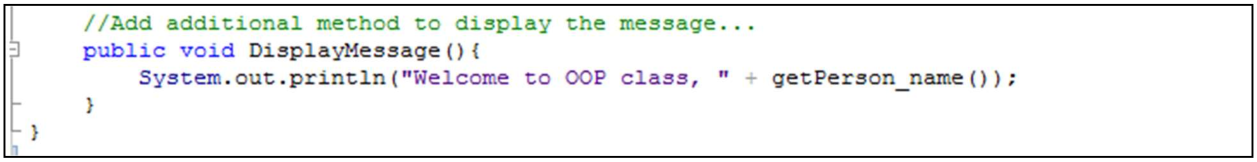
```
    //Define our own constructor...
    public Welcome(String person) {
        //Assigned the person name
        setPerson_name(person);
    }
```

Step 8: Define getter and setter for each of the instance variable.



```
    //Define getter and setter for instance variables...
    public void setPerson_name(String person_name) {
        this.person_name = person_name;
    }

    public String getPerson_name() {
        return (this.person_name);
    }
```

Step 9: Define additional method to display the message.



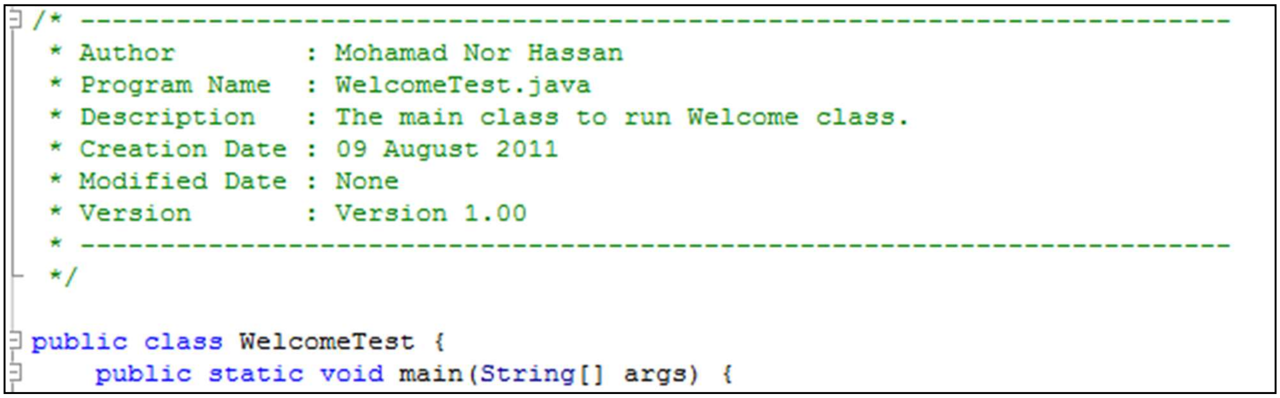
```
    //Add additional method to display the message...
    public void DisplayMessage(){
        System.out.println("Welcome to OOP class, " + getPerson_name());
    }
}
```

Step 10: Save the file in sub-working folder called Activity 1.

Step 11: Compile the program.

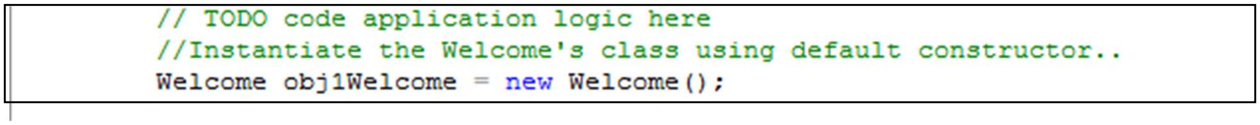
Step 12: Create new file/class called WelcomeTest.java

Step 13: Define the WelcomeTest class as below:



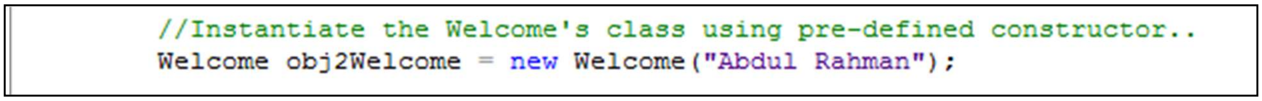
```
/* -----------------------------------------------------------------------
 * Author        : Mohamad Nor Hassan
 * Program Name  : WelcomeTest.java
 * Description   : The main class to run Welcome class.
 * Creation Date : 09 August 2011
 * Modified Date : None
 * Version       : Version 1.00
 * -----------------------------------------------------------------------
 */

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

Step 14: Instantiate Welcome class using default constructor.



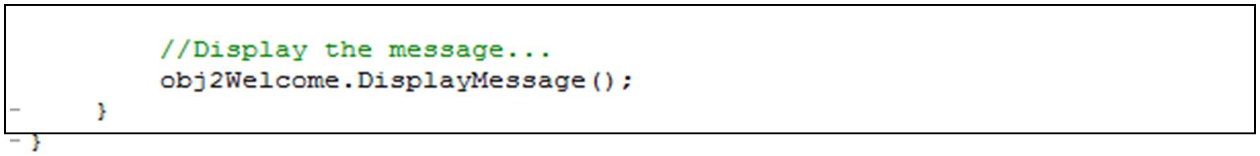
```
        // TODO code application logic here
        //Instantiate the Welcome's class using default constructor..
        Welcome obj1Welcome = new Welcome();
```

Step 15: Instantiate Welcome class using 2nd constructor:



```
        //Instantiate the Welcome's class using pre-defined constructor..
        Welcome obj2Welcome = new Welcome("Abdul Rahman");
```

Step 16: Display the message by invoke the method *DisplayMessage()*.



```
        //Display the message...
        obj2Welcome.DisplayMessage();
    }
}
```

Step 17: Compile the program.

Step 18: Run and evaluate the output.

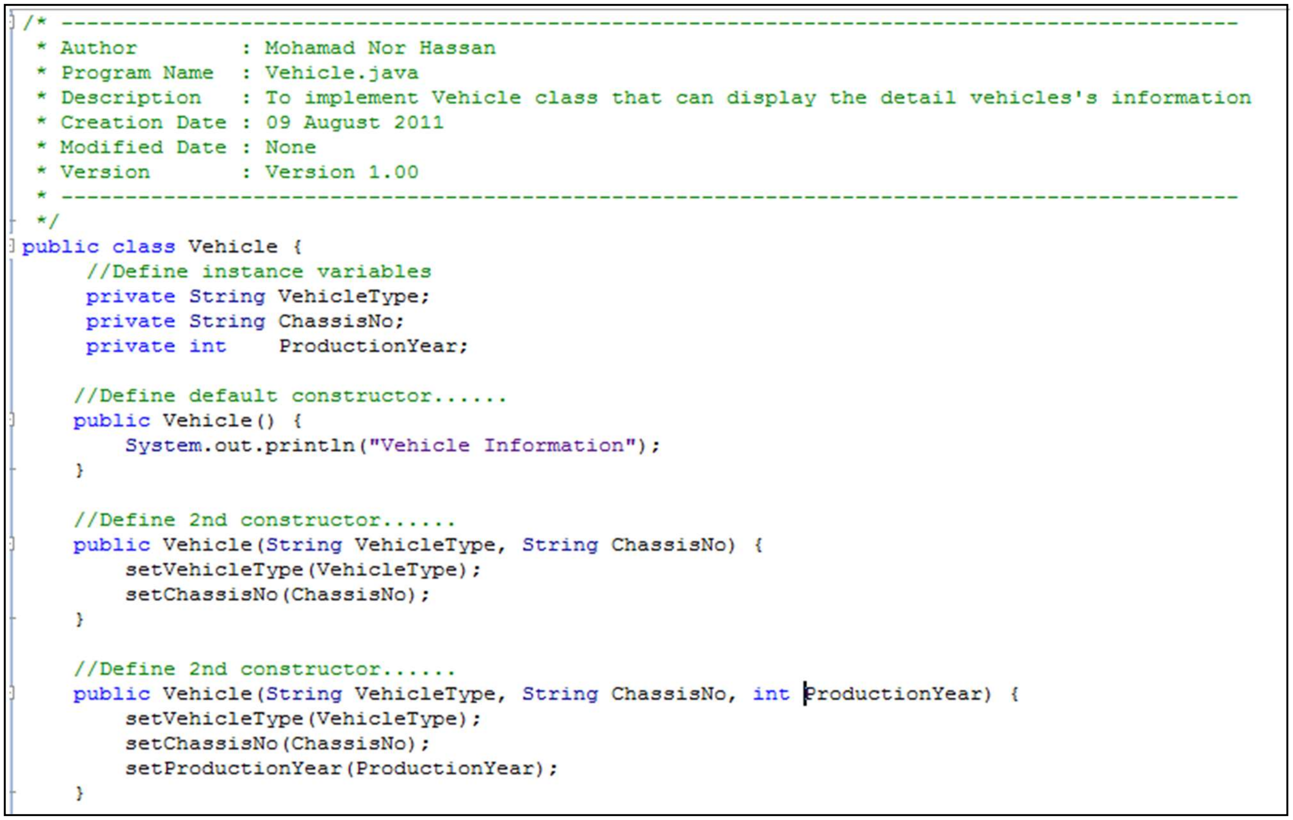
2.2 Activity 2

2.2.1 Objective

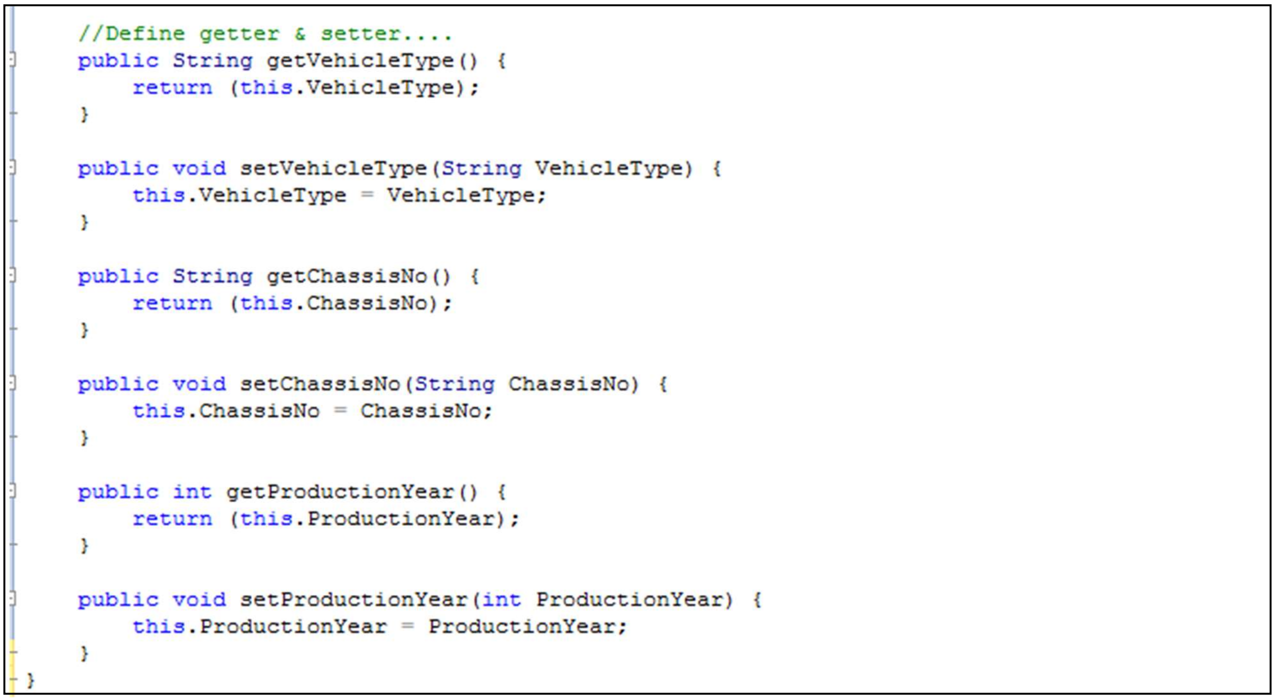
How to instantiate an object of the class.

2.2.2 Problem Description

Vehicle’s class shows in Figure 1 will be used to display details information for specific vehicle. Create main program and instantiate each constructor, and display the information for each of the attribute.



```
/* -------------------------------------------------------------------------------------------
 * Author        : Mohamad Nor Hassan
 * Program Name  : Vehicle.java
 * Description   : To implement Vehicle class that can display the detail vehicles's information
 * Creation Date : 09 August 2011
 * Modified Date : None
 * Version       : Version 1.00
 * -------------------------------------------------------------------------------------------
 */
public class Vehicle {
     //Define instance variables
     private String VehicleType;
     private String ChassisNo;
     private int    ProductionYear;

    //Define default constructor......
    public Vehicle() {
        System.out.println("Vehicle Information");
    }

    //Define 2nd constructor......
    public Vehicle(String VehicleType, String ChassisNo) {
        setVehicleType(VehicleType);
        setChassisNo(ChassisNo);
    }

    //Define 2nd constructor......
    public Vehicle(String VehicleType, String ChassisNo, int ProductionYear) {
        setVehicleType(VehicleType);
        setChassisNo(ChassisNo);
        setProductionYear(ProductionYear);
    }
```



```
    //Define getter & setter....
    public String getVehicleType() {
        return (this.VehicleType);
    }

    public void setVehicleType(String VehicleType) {
        this.VehicleType = VehicleType;
    }

    public String getChassisNo() {
        return (this.ChassisNo);
    }

    public void setChassisNo(String ChassisNo) {
        this.ChassisNo = ChassisNo;
    }

    public int getProductionYear() {
        return (this.ProductionYear);
    }

    public void setProductionYear(int ProductionYear) {
        this.ProductionYear = ProductionYear;
    }
}
```

Figure 1- Vehicle class

[Estimated Time: 15 minutes]

2.3 Activity 3

2.3.1 Objective

Initialize the constructor and introduce getter and setter methods.

2.3.2 Problem Description

The *Student()* class consists of attribute such as student id, name, and status. The status can be assigned either active or suspend. Your program must have minimum three (3) constructors, that will behave differently when the program is executed at runtime. Create the main program and display the student name, course and his/her status.

[Estimated Time: 35 minutes]

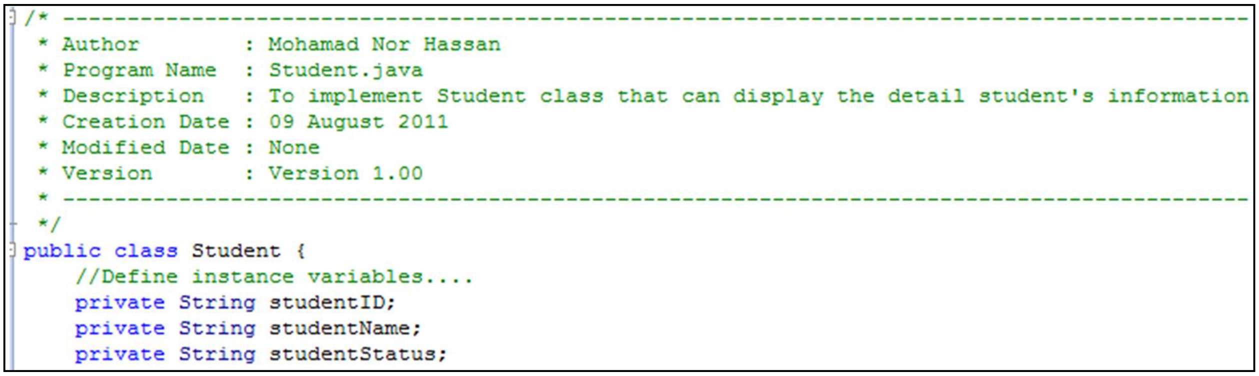
2.3.3 Solution Activity 3

Step 1: Go to CSF3043 –> Lab – Chapter 2's folder and create Sub-working folder called Activity 3.

Step 2: Open IDE Netbeans or JCreator.

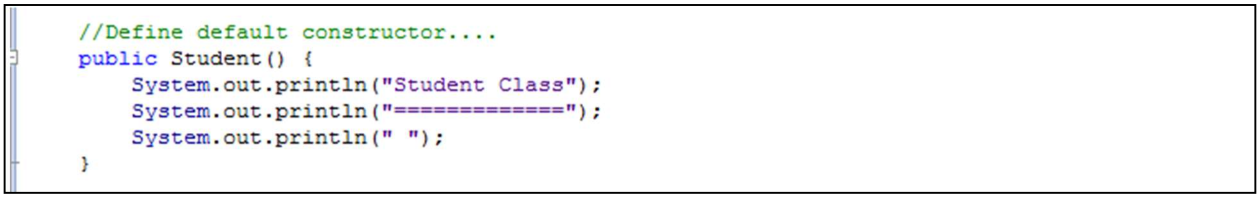
Step 3: Create new file/class called Student.java.

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



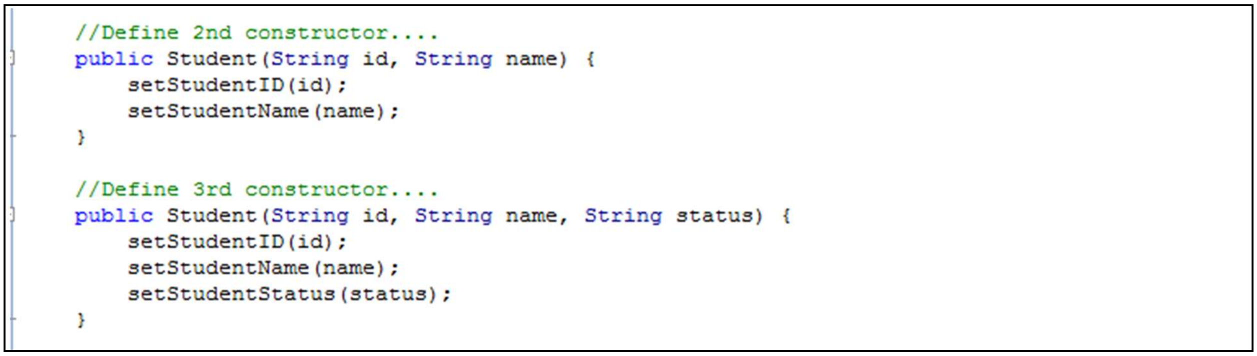
```
/* -------------------------------------------------------------------------------------------
 * Author        : Mohamad Nor Hassan
 * Program Name  : Student.java
 * Description   : To implement Student class that can display the detail student's information
 * Creation Date : 09 August 2011
 * Modified Date : None
 * Version       : Version 1.00
 * -------------------------------------------------------------------------------------------
 */
public class Student {
    //Define instance variables....
    private String studentID;
    private String studentName;
    private String studentStatus;
```

Step 5: Define the default constructor for Student's class.



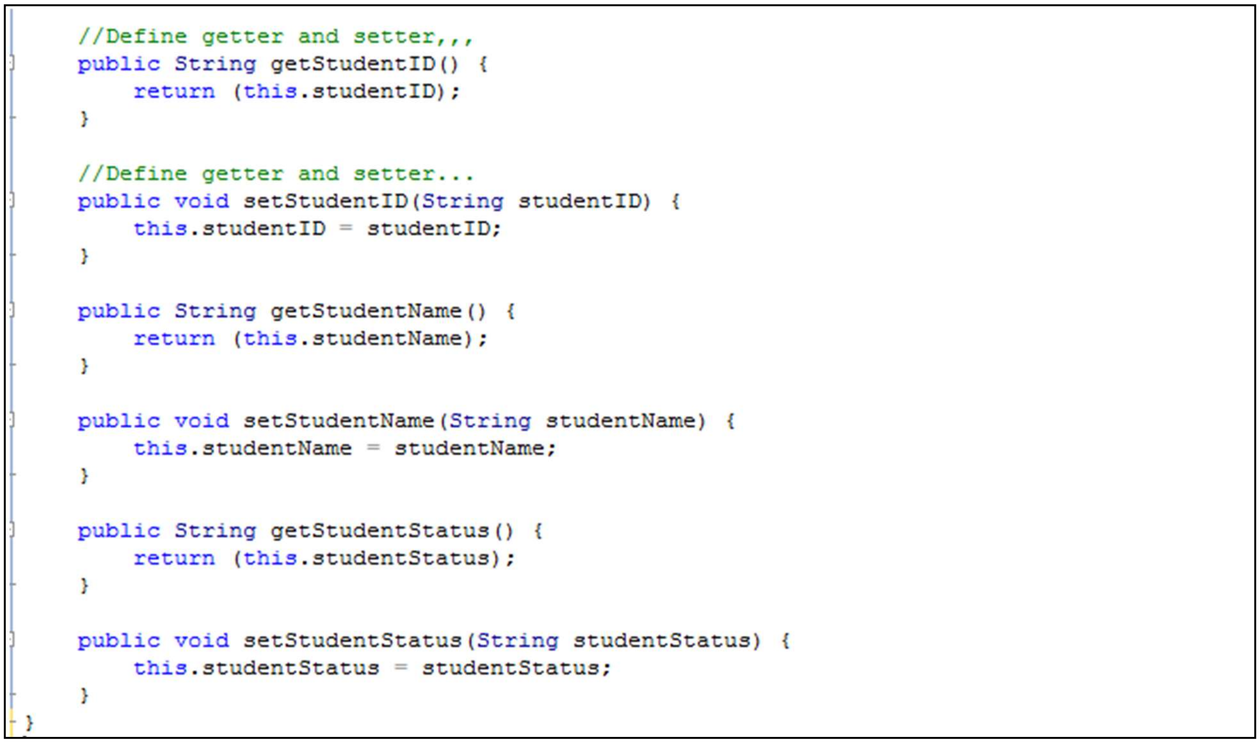
```
    //Define default constructor....
    public Student() {
        System.out.println("Student Class");
        System.out.println("=============");
        System.out.println(" ");
    }
```

Step 6: Define 2nd and 3rd constructor for Student class.



```
    //Define 2nd constructor....
    public Student(String id, String name) {
        setStudentID(id);
        setStudentName(name);
    }

    //Define 3rd constructor....
    public Student(String id, String name, String status) {
        setStudentID(id);
        setStudentName(name);
        setStudentStatus(status);
    }
```

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



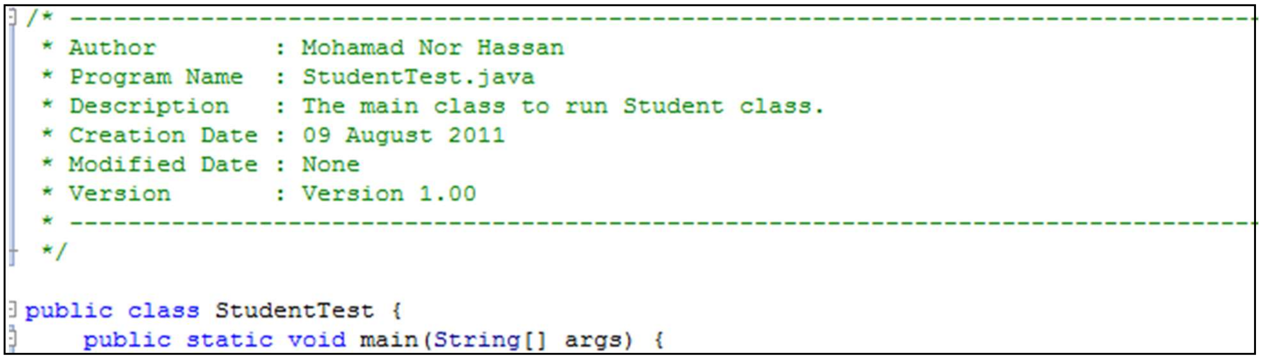
```
    //Define getter and setter,,,
    public String getStudentID() {
        return (this.studentID);
    }

    //Define getter and setter...
    public void setStudentID(String studentID) {
        this.studentID = studentID;
    }

    public String getStudentName() {
        return (this.studentName);
    }

    public void setStudentName(String studentName) {
        this.studentName = studentName;
    }

    public String getStudentStatus() {
        return (this.studentStatus);
    }

    public void setStudentStatus(String studentStatus) {
        this.studentStatus = studentStatus;
    }
}
```

Step 8: Save the file in sub-working folder called Activity 3.

Step 9: Compile the program.

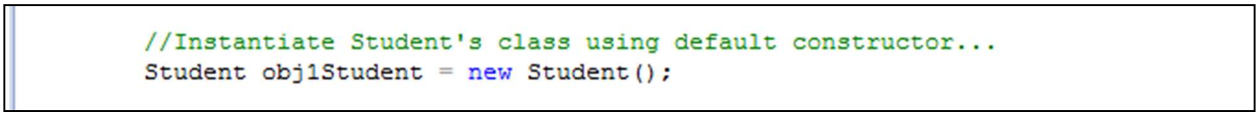
Step 10: Create new file/class called StudentTest.java

Step 11: Define the StudentTest class as below



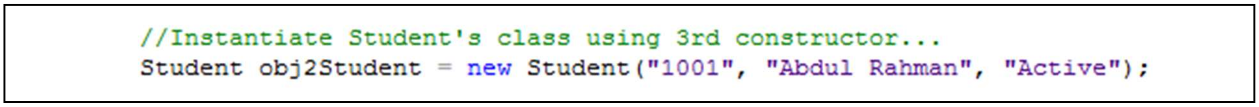
```
/* ---------------------------------------------------------------------------------
 * Author         : Mohamad Nor Hassan
 * Program Name   : StudentTest.java
 * Description    : The main class to run Student class.
 * Creation Date : 09 August 2011
 * Modified Date : None
 * Version        : Version 1.00
 * ---------------------------------------------------------------------------------
 */

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

Step 12: Instantiate Student class using default constructor.



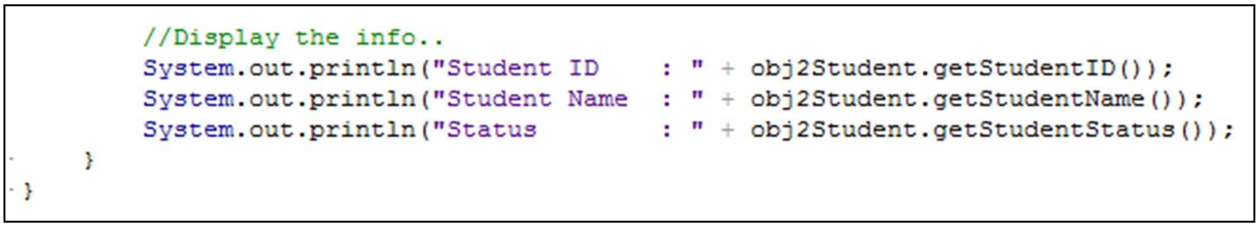
```
        //Instantiate Student's class using default constructor...
        Student obj1Student = new Student();
```

Step 13: Instantiate Student class using 3rd constructor.



```
        //Instantiate Student's class using 3rd constructor...
        Student obj2Student = new Student("1001", "Abdul Rahman", "Active");
```

Step 14: Display the student name, course and his/her status.



```
        //Display the info..
        System.out.println("Student ID    : " + obj2Student.getStudentID());
        System.out.println("Student Name  : " + obj2Student.getStudentName());
        System.out.println("Status        : " + obj2Student.getStudentStatus());
    }
}
```

Step 14: Compile the program.

Step 15: Run and evaluate the output.

2.4 Activity 4

2.4.1 Objective

Using primitive data type in class.

2.4.2 Problem Description

Write a program to convert string number into integer, integer number into double. Prompt the user at main program to enter two (2) numbers that representing string number and integer number. The program should validate the string input key-in by user is number. If the input is string (a, b, c, etc.), then assigned the input as “0”. Otherwise, display the error message. Finally, display the out of conversion.

[Estimated Time: 40 minutes]

Lab Exercises

Objective

i. Writing a small object oriented program using attributes, constructor and methods with primitive data type.

ii. To passing a parameters and details implementation of constructor in computing ticket sales of Science Exhibition.

Problem Description

You need to write the program to calculate the total ticket sales of a concert. There are three types of seating: A, B and C. The program must accept the number of tickets sold and the price of a ticket for each of the three types of seats. The total sales are computed as follow:

```
totalSales =  numberofA_Seats * pricePerA_Seat +
              numberofB_Seats * pricePerB_Seat +
              numberofC_Seats * pricePerC_Seat
```

Write a class called Ticket Sale and use a class called TicketSaleTest to run the program and display total sales for ticket A, B and C. Finally display total sales of the ticket

[Estimated Time: 50 minutes]