

LAB 3 – Advanced to Class and Object – Part 1

Objectives:

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

- Use a reference data type and understand the composition/aggregation used in the class.
- Apply and used appropriate access modifier inside the class.
- Use reference this and overloading constructor.

3.1 Activity 1

3.1.1 Objective

To understand the use of composition (used reference data type) for reusing an existing class in other classes.

3.1.2 Problem Description

- Given the employee composed of the class called Date, employeeid, name and hiredate.
- The class Date is given below:

```
public class Date
{
    private int month; // 1-12
    private int day;   // 1-31 based on month
    private int year;  // > 0

    // constructor: call checkMonth to confirm proper value for month;
    // call checkDay to confirm proper value for day
    public Date( int theMonth, int theDay, int theYear )
    {
        month = checkMonth( theMonth ); // validate month
        year = checkYear( theYear ); // validate year
        day = checkDay( theDay ); // validate day

        System.out.printf(
            "Date object constructor for date %s\n", toString() );
    }

    // utility method to confirm proper year value
    private int checkYear( int testYear )
    {
        if ( testYear > 0 ) // validate year
            return testYear;
        else // day is invalid
        {
            System.out.printf(
                "Invalid year (%d) set to 1.\n", testYear );
            return 1;
        } // end else
    } // end method checkYear
}
```

```

1 private int checkMonth( int testMonth )
  {
    if ( testMonth > 0 && testMonth <= 12 ) // validate month
      return testMonth;
    else // month is invalid
    {
      System.out.printf(
        "Invalid month (%d) set to 1.\n", testMonth );
      return 1; // maintain object in consistent state
    } // end else
  } // end method checkMonth

  // utility method to confirm proper day value based on month and year
1 private int checkDay( int testDay )
  {
    int daysPerMonth[] =
      { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

    // check if day in range for month
    if ( testDay > 0 && testDay <= daysPerMonth[ month ] )
      return testDay;

    // check for leap year
    if ( month == 2 && testDay == 29 && ( year % 400 == 0 ||
      ( year % 4 == 0 && year % 100 != 0 ) ) )
      return testDay;

    System.out.printf( "Invalid day (%d) set to 1.\n", testDay );

    return 1; // maintain object in consistent state
  }

  // return a String of the form month/day/year
1 public String toString()
  {
    return String.format( "%d/%d/%d", month, day, year );
  }
}

```

3. Copy the class Date and implement a class Employee. Finally, use main program to display the employee id, name and hire date using format day/month/year.

[Estimated Time: 45 minutes]

3.1.3 Solution Activity 1

Step 1: Go to CSE3101 folder and create sub-folder called as Lab – Chapter 3 Part 1.

Step 2: Go to CSE3101 → Lab – Chapter 3 Part 1 folder and create sub-working folder called Activity 1.

Step 3: Open IDE Netbeans or JCreator.

Step 4: Create and compile new file/class called Date.java.

Step 5: Create new class called Employee.java.

Step 6: Define the class and instance variables as below:

```
1  /*
2  * Author      : Mohamad Nor Hassan
3  * Program Name : Employee.java
4  * Description  : To implement Employee class that can display the detail vehicles's information
5  * Creation Date : 09 August 2011
6  * Modified Date : None
7  * Version     : Version 1.00
8  * -----
9  */
10 public class Employee {
11
12     private String employeeid;
13     private String name;
14     Date    hiredate;
15 }
```

Step 7: Define the constructor for Employee's class.

```
1  //Define constructor...
2  public Employee(String employeeid, String name, Date hiredate) {
3      setEmployeeid(employeeid);
4      setName(name);
5      setHiredate(hiredate);
6  }
7  }
```

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

```
//Define getter & setter....  
public void setEmployeeid(String employeeid) {  
    this.employeeid = employeeid;  
}  
  
public void setName(String name) {  
    this.name = name;  
}  
  
public void setHiredDate(Date hiredDate) {  
    this.hiredDate = hiredDate;  
}  
  
public String getEmployeeid() {  
    return (this.employeeid);  
}  
  
public String getName() {  
    return (this.name);  
}  
  
public Date getHiredDate() {  
    return (this.hiredDate);  
}
```

Step 9: Define additional method to display the message.

```
//Display the output..  
public String toString(){  
    return String.format("Employee Id: %s, Name: %s, Hired Date: %s ", getEmployeeid(), getName(), getHiredDate());  
}  
}
```

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

Step 11: Compile the program.

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

Step 13: Define EmployeeTest class as below:

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

Step 14: Instantiate Date class.

```
    */  
    public static void main(String[] args) {  
        // TODO code application logic here  
  
        //Define hiredate...  
        Date hire = new Date(01, 05, 2010);  
    }
```

Step 15: Instantiate Employee's class and display the message by invoking the constructor.

```
        //Create and instantiate object for class Employee()...  
        Employee employee = new Employee("1005", "Thomas Wu", hire);  
        System.out.println(employee);  
    }  
}
```

Step 16: Compile the program.

Step 17: Run and evaluate the output.

3.2 Activity 2

3.2.1 Objective

To demonstrate the use of access modifier private, public to member of the class and non-member of the class.

3.2.2 Problem Description

You need to implement a Rectangle class that has the requirements below:

- i. Have a variables width and height. The variables can only be accessed by their member's class. Additional variable for area is declared for member and non-member class.
- ii. Have an additional private method to calculate the area of the rectangle.
- iii. Write a simple class Rectangle.
- iv. Write a program to test class Rectangle. Try to use the variable height and width and assign the value 4 and 2 respectively. In addition using variable area to calculate area in main program.
- v. Compile the program Rectangle and main program. Briefly explain your result.
- vi. If there is an errors, how to fix it. Why the errors occur during the compilation?

[Estimated Times: 40 minutes]

3.3 Activity 3

3.3.1 Objective

Using reference *this* to current object and overloading constructor.

3.3.2 Problem Description

Write a program to convert the distance in miles to kilometer. Your program should accept the input from the user. Finally, create the main program to display the distance before and after conversion takes place. Use this qualifier to differentiate the assignment of each of variable.

Note: 1 mile = 1.609344 km

[Estimated Time: 35 minutes]

Lab Exercises

Objective

Writing a simple program based on specific case study by applying the concept access modifier, reference data type, reference this and using overloading constructor.

Problem Description

You need to write the program to display bank transaction. Transaction class consists of these attributes:

Account number - can be accessed by other program via specific method.

Account name - can be accessed by other program via specific method.

Type of account - can be access by any program.

Transaction date - can be accessed by other program via specific method.

The transaction date has a Date class. Write the main program to display:

- i. The transaction for account = "10021", account holder = "Abdul Rahman"
- ii. The transaction for account = "10021", account holder = "Abdul Rahman" and transaction date = "12 Aug 2011"

[Estimated Time: 50 minutes]