LAB 4 – Advanced to Class and Object – Part 2

Objectives:

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

i. Differentiate and how use static, and final instance for specific problem.
ii. Using the enumeration as a systematic structure to represent a list of constant.
iii. Create and use a package.
iv. Using methods in Java class object and other classes such as *ArrayList()*.

4.1 Activity 1

4.1.1 Objective

Writing a class with static field/instance variable data type.

4.1.2 Problem Description

Write a program to display person in-charge, transaction type (D-Deposit, W-Withdrawal), transaction number and balance when the user withdrawing or deposit the money from the specific account.

You must define transaction number and balance as a static instance variable and each of the variables defaulted to 0 and 1000.00 respectively.

Create the main program and run the following transactions:

1st transaction as a withdrawal transaction. Pass the parameters to the object using the following information; account number = 1002, account name = "Ali", transaction as withdrawal ("W") and the withdrawal amount is RM400.00.

2nd transaction as a deposit transaction. Pass the parameters to the object using the following information; account number = 1002, account name = "Ali", transaction as deposit ("D") and the deposit amount is RM100.00.

3rd transaction as a withdrawal transaction. Pass the parameters to the object using the following information; account number = 1002, account name = "Ali", transaction as withdrawal ("W") and the withdrawal amount is RM600.00

What are the last transaction number and the current balance?

[Estimated Time: 25 minutes]

4.1.3 Solution Activity 1

Algorithms

Step 1 – Declare a class Account and attributes.
Step 2 – Assigned value to static attribute as 0 and 1000.00
Step 3 – Create constructor
Step 4 – Define getter and setter for each attributes
Step 5 – Add additional operation to display final result/output.
Step 6 – Implement logic in constructor
Step 6.1 – Perform increment for each of transaction
Step 6.2 – Initialize the value for Person In-Charge and Type of transaction
Step 6.3 – If transaction type = 'D', add the current balance.
Step 6.4 – If transaction type = 'W', deduct the current balance.
Step 7 - Declare a class AccountTransaction.
Step 8 – Instantiate class Account and display the result of each of the requirement.

Programming

Step 1: Go to CSF3043 folder and create sub-folder called as Lab – Chapter 3 Part 2.

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

Step 3: Open IDE Netbeans or JCreator.

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

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

```
/* ---------------------------------------------------------------------------
 * Author         : Mohamad Nor Hassan
 * Program Name   : Account.java
 * Description    : To implement the business requirements for customer's account.
 * Creation Date  : 22 Jan 2011
 * Modified Date  : None
 * Version        : Version 1.00
 * ---------------------------------------------------------------------------
 */
public class Account {

    //Define instance variables....
    private String  PersonInCharge;
    private String  Transactiontype;
    private double  withdraw;
    private double  debit;

    //Define static instance variables and default to RM1000.00 ....
    private static double  balance = 1000.00;

    //Define static instance variables and default to 0 ....
    private static int     transactionNo = 0;
```

Step 6: Define the constructor for Account class.

```
    public Account() {
    }

    //Define 2nd constructor....
    public Account(String PersonInCharge,
                   String transaction, double amount) {

        //Generate new transaction no...
        transactionNo++;

        //initialize the value...
        setPersonInCharge(PersonInCharge);
        setTransactiontype(transaction);

        //Identify the transaction.... D = Deposit... W = Withdrawal....
        if (getTransactiontype().equals("D")){
            setBalance( getBalance() + amount );
        }
        else if (getTransactiontype().equals("W")){
            setBalance( getBalance() - amount );
        }
    }
```

Step 7: Define getter and setter for each of the instance variable.

```
    //Define setter and getter..........
    public String getPersonInCharge() {
        return (this.PersonInCharge);
    }

    public void setPersonInCharge(String PersonInCharge) {
        this.PersonInCharge = PersonInCharge;
    }

    public String getTransactiontype() {
        return (this.Transactiontype);
    }

    public void setTransactiontype(String Transactiontype) {
        this.Transactiontype = Transactiontype;
    }
```

Step 8: Define getter and setter for each static instance variable

```
    //Define getter and setter static variables...
    public static double getBalance() {
        return (Account.balance);
    }

    public static void setBalance(double balance) {
        Account.balance = balance;
    }

    public static int gettransactionNo(){
        return (Account.transactionNo);
    }

    public static void settransactionNo(int transactionNo) {
        Account.transactionNo = transactionNo;
    }
```

Step 9: Define additional method to display the message.

```
    public void displayBalance(){
        System.out.println("--------------------------");
        System.out.printf("Person In-Charge  = %s\n", getPersonInCharge());
        System.out.printf("Transaction No    = %d\n", gettransactionNo());
        System.out.printf("Current balance   = %.2f\n", getBalance());
        System.out.println("           ");
    }
}
```

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

Step 11: Compile the program.

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

Step 13: Define the AccountTransaction class as below:

```
/* ------------------------------------------------------------------
 * Author         : Mohamad Nor Hassan
 * Program Name   : AccountTransaction
 * Description    : To execute class Account()
 * Creation Date  : 22 Jan 2011
 * Modified Date  : None
 * Version        : Version 1.00
 * ------------------------------------------------------------------
 */
public class AccountTransaction {
    public static void main(String[] args) {
        // TODO code application logic here
```

Step 14: Instantiate Account class using 2nd constructor and display the output.

```
        //Task 1 -> Define the 1st transaction as withdrawal.....
        //Instantiate the Account()'s object as account1....
          Account account1 = new Account("Ali", "W", 400.00);

        //Display the current balance...
          account1.displayBalance();

        //Task 2 -> Define the 2nd transaction as deposit.....
        //Instantiate the Account()'s object as account2....
          Account account2 = new Account("Ali", "D", 100.00);

        //Display the current balance...
          account2.displayBalance();

        //Task 3 -> Define the 3rd transaction as wihtdrawal.....
        //Instantiate the Account()'s object as account3....
          Account account3 = new Account("Ali", "W", 600.00);

        //Display the current balance...
          account3.displayBalance();
    }
}
```

Step 15: Compile the program.

Step 16: Run and evaluate the output.

4.2 Activity 2

4.2.1 Objective

Writing a class with final field/instance variable data type.

4.2.2 Problem Description

Write a program to convert the currency from Malaysian Ringgit (MYR) to Pounds Sterling (£). The rate is 1£ is equivalent to MYR4.320. Your program should accept the input MYR key-in by the user. Finally, display the conversion.

[Estimated Time: 25 minutes]

4.3 Activity 3

4.3.1 Objective

Implement enumeration concept for listing of subject.

4.3.2 Problem Description

Create an enumeration called as Subject

Details of enum declaration consist of following information:
1. Enum constant -> Arguments
2. CSE3501 -> ("Software Engineering Basic", "BCS(SE)")
3. CSE3502 -> ("Software Requirements Engineering", "BCS(SE)") CSE3503 -> ("Software Architecture", "BCS(SE)")
4. CSE3504 -> ("Software Testing", "BCS(SE)")
5. MMM3401->("Principle of Maritime Management", "BSC(IM)")
6. CSA3501->("Maritime Informatics Technology and Application", "BSC(IM)")

The constructor for enum Subject has a signature of subject name and course name

The enum Subject has a getter for subject name and course name

For testing purposes, create new main class known as *SubjectEnumTest()* and write the following implementation to test the enum Subject:
1. Display all the subjects.
2. Display the subject between subject code CSE3501 to CSE3503.
3. Display the subject code CSE3502 and CSE3504.

[Estimated Time: 25 minutes]

4.4 Activity 4

4.4.1 Objective

Create and use the package.

4.4.2 Problem Description

Create a program for the user to key-in their name and date of birth. In your program, create a package using the keyword package and use the package via import statement.

6.4.1 Solution Activity 4

Step 1: Go to CSF3043 folder and create sub-folder called as Lab – Chapter 3 Part 2.

Step 2: Go to CSF3043 –> Lab – Chapter 3 Part 2 folder and create Sub-working folder called Activity 4.

Step 3: Open IDE Netbeans or JCreator.

Step 4: Create a new package called PackageUtility.

Step 5: Create new class called Date.java in PackageUtility. Use existing class Date as defined in Lab 3.

Step 6: Create another package called PackageFTKKI.

Step 7: Create a new class called Person.java in PackageFTKKI.

Step 8: Define the instance variables as follow: private String name; private Date dateOfBirth;

Step 9: Define the constructor for class Person.

Step 10: Define setter and getter methods for each of instance variable.

Step 11: Override *toString()* method to return name and date of birth value.

Step 12: Define class PersonTest in PackageFTKKI.

Step 13: Define statement to read input key-in by user.

Step 14: Instantiate Person class and display the output based on the input key-in by user.

Step 15: Compile the program. If errors, identify and fix the errors.

Step 16: Run and evaluate the output

[Estimated Time: 25 minutes]

4.5 Activity 5

4.5.1 Objective

Using *toString()* method and *ArrayList()* class.

4.5.2 Problem Description

Create a program to store the two (2) collection of String object using an *ArrayList()* class. In your main program, pass two (2) collection of String via constructor. Finally, display each of the collection String by invoking *toString()* method.

[Estimated Time: 30 minutes]

4.5.3 Solution Activity 5

Algorithms

Step 1 – Declare a class StringCollection and attributes.
Step 2 – Instantiate the ArrayList class.
Step 3 – Create constructor and passed two String parameters.
Step 4 – Initialize constructor and add two String parameters to object ArrayList.
Step 5 – Define getter and setter for each attributes
Step 6 – Add *toString()* operation to display final result/output.
Step 7 – Implement logic in *toString()*
Step 7.1 – Declare local variables as String datatype
Step 7.2 – Perform looping for ArrayList object and concatenate to local variable
Step 7.3 – Return the result using *String.format*. Step 7 - Declare a class StringCollection.
Step 8 – Instantiate class Account and passing two (2) String parameters. Step 8 – Display the result.

Programming

Step 1: Go to CSF3043 –> Lab – Chapter 3 Part 2 folder and create Sub-working folder called Activity 5.

Step 2: Open IDE Netbeans or JCreator.

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

Step 3: Import *java.util.ArrayList*, define the class and instance variables as below:

```
/* ---------------------------------------------------------------------------
 * Author         : Mohamad Nor Hassan
 * Program Name   : StringCollection.java
 * Description    : To implement the collection of string by using ArrayList()
 * Creation Date  : 13 Aug 2011
 * Modified Date  : None
 * Version        : Version 1.00
 * ---------------------------------------------------------------------------
 */
import java.util.ArrayList;

public class StringCollection {

    //Define instance variables...
    private String str1;
    private String str2;
```

Step 4: Instantiate ArrayList class as below:

```
    //Instantiate the class Array list...
    ArrayList<String> objStringCollection = new ArrayList<String>();
```

Step 5: Define the constructor for StringCollection class.

```
    public StringCollection(String str1, String str2) {
        //Initiliaze the string values...
        setStr1(str1);
        setStr2(str2);

        //Add the value into ArrayList...
        objStringCollection.add(getStr1());
        objStringCollection.add(getStr2());
    }
```

Step 6: Define getter and setter for each attributes.

```
    //Define getter and setter...
    public String getStr1() {
        return (this.str1);
    }

    public void setStr1(String str1) {
        this.str1 = str1;
    }

    public String getStr2() {
        return (this.str2);
    }

    public void setStr2(String str2) {
        this.str2 = str2;
    }
```

Step 7: Define the details implementation of *toString()* method.

```
    public String toString(){
        //To display the value of ArrayList...
        String result = "";
        for ( String s : objStringCollection )
        {
            result += s + " ";
        }

        return String.format("%s  =  %s\n",
                             "Collection of ArrayList ", result);
    }
}
```

Step 8: Save the file in sub-working folder called Activity 5.

Step 9: Compile the program.

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

Step 11: Define the class StringCollectionTest by passing two (2) String values "M" and "N". Finally, display the output by invoking the constructor via *System.out.println()*.

```
/* ---------------------------------------------------------------------------
 * Author         : Mohamad Nor Hassan
 * Program Name   : StringCollectionTest.java
 * Description    : To execute class StringCollection()
 * Creation Date  : 13 Aug 2011
 * Modified Date  : None
 * Version        : Version 1.00
 * ---------------------------------------------------------------------------
 */
public class StringCollectionTest {
    public static void main(String[] args) {
        // TODO code application logic here
        //Instantiate the class StringCollection()...
        StringCollection objStringCollection = new StringCollection("M", "N");

        //Display the collection...
        System.out.println(objStringCollection);
    }
}
```

Lab Exercises

Objective:

Writing an object-oriented program solution by applying the concept students learned in Lab 4.

Problem Description:

Customer who buying the books will entitle for specified discount. Customer, who has a membership, will receive 15% discount. There are two (2) types of a book. For beginner level, the price is RM50.00 and RM100.00 for advanced level.

The program must have three (3) constructors, including default constructor. The second constructor is used to find the price per book based on the type of book key-in by the user.

The third constructor is used to calculate the total amount customer needs to pay after he/she key-in the information such as membership (Y/N), type of book (A-advanced, B-beginner) and quantity of a book bought.

You need to write the program called *BookSales.java* and using the main class to display the price for each type of the book and display the quantity of the book customer bought and total amount customer needed to pay.

[Estimated Time: 50 minutes]