

Lab Exercise for Week 09

PROGRAMMING WITH THE JAVA COLLECTION

BITP 3113 Object-Oriented Programming

Author

Emaliana Kasmuri

Fakulti Teknologi Maklumat dan Komunikasi

Universiti Teknikal Malaysia Melaka

Table of Contents

Learning Outcomes	1
Tools	1
Supporting Materials	1
Exercise 1	2
Exercise 2	2
Exercise 3	3
Exercise 4	3
Exercise 5	4
Exercise 6	4
Exercise 7	5
Exercise 8	6

Learning Outcomes

At the end of this lab exercise, the student should be able to:-

1. Implement object-oriented principles to construct Java classes that represent real-world entities and their relationships,
2. Utilize Java Collection framework classes, including `ArrayList`, `Map`, and `LinkedHashMap`, to manage data effectively within Java.
3. Employ inheritance and method overriding to design a modular and adaptable class hierarchy that accommodates variations of business process.

Tools

The exercise for this lab session is using the following tools:-

1. Eclipse for Java Developers

Supporting Materials

The reference files and supporting materials are available on the ulearn course site.

Exercise 1

The instruction to execute this exercise are as follows:-

- Download NetWorthArrayApp.zip from ulearn.
- Import into Eclipse.
- Execute the application.
- Study and observe the program.

Exercise 2

Christian Dior is one of the brands under the LVMH (Louis Vuitton Moët Hennessy) umbrella. It is a luxury brand that has offers various categories of product including fashion, leather goods, perfumes, cosmetics, watches, and jewelry. The brand is available worldwide including Malaysia. Christian Dior Malaysia need to report its sales to the LVMH by the end of this month. The sales data are described in Table 1.

Table 1: Christian Dior Malaysia sales data

Item Name	Price Per Unit (RM)	Quantity Sold
Medium Lady Dior Bag - Black Cannage Lambskin	31,500.00	56
Small Dior Book Tote - Blue Dior Oblique Embroidery and Calfskin (26.5 x 22 x 14 cm)	17,000.00	43
Saddle Bag with Strap - Golden Saddle Grained Calfskin	20,500.00	67
Small Dior Toujours Vertical Tote Bag - Gray Macrocannage Calfskin	15,500.00	15
30 Montaigne East-West Bag with Chain - Black Calfskin	15,500.00	26

- Define a Java class to represent each item in Table 1.
- Define another class to calculate total sales of each item described in Table 1.
- Define a class with a `main()` method to store the sales data in Table 1 in an array object.
- The `main()` method shall display the sales amount of each item and total sales of all items using the array defined in the previous questions.
- All Java codes must be neatly written, comply to the Java coding standard with sufficient and descriptive comments.
- Compress the source codes (only .java files) and an output screen shot in one.zip file.

- g. Submit the file in ulearn.

Exercise 3

- a. Change the implementation of the previous question from the conventional array to ArrayList.
- b. Then add the additional data in All Java codes must be neatly written, comply to the Java coding standard with sufficient and descriptive comments.
- c. Compress the source codes (only .java files) and an output screen shot in one.zip file.
- d. Submit the file in ulearn.
- e. Table 2 and display it after the initial implementation .
- f. All Java codes must be neatly written, comply to the Java coding standard with sufficient and descriptive comments.
- g. Compress the source codes (only .java files) and an output screen shot in one.zip file.
- h. Submit the file in ulearn.

Table 2: Additional sales data

Item Name	Price Per Unit (RM)	Quantity Sold
Medium Dior Book Tote - Multicolor Dior Pivoine Embroidery (36 x 27.5 x 16.5 cm)	16,000.00	24
Small Lady Dior My ABCDior Bag - Blush Pink Cannage Lambskin	29,500.00	8

Exercise 4

The instruction to execute this exercise are as follows:-

- a. Download NetWorthHashMap.zip from ulearn.
- b. Import into Eclipse.
- c. Execute the application.
- d. Study and observe the program.

Exercise 5

- i. Create a Java program to store the information in Table 1 and All Java codes must be neatly written, comply to the Java coding standard with sufficient and descriptive comments.
- j. Compress the source codes (only .java files) and an output screen shot in one.zip file.
- k. Submit the file in ulearn.

Table 2 using [Map](#) and [LinkedHashMap](#) using the knowledge and skill comprehended from the previous exercises. The program shall include the following:-

- a. Display all the keys used in the map. Used the [keyset\(\)](#) method.
- b. Demonstrate object searching using specified key and display its result.
- c. All Java codes must be neatly written, comply to the Java coding standard with sufficient and descriptive comments.
- d. Compress the source codes (only .java files) and an output screen shot in one.zip file.
- e. Submit the file in ulearn.

Exercise 6

In Malaysia, all individuals who earn income must register as a taxpayer. The UML class in Figure 1 shows a base class **TaxPayer** that stores basic taxpayer information.

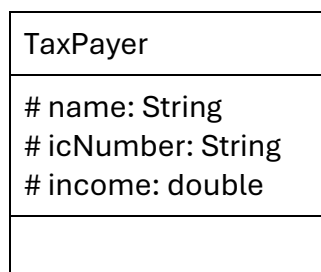


Figure 1: Class TaxPayer in UML notation

- a. The class definition in Figure 1 is incomplete. Add the following requirement to complete the **TaxPayer** class in Figure 1.
 - a. Add a constructor to initialize all attributes.
 - b. Add the getter and setter methods for all attributes.
- b. Implement the **TaxPayer** class using Java.
- c. Design a UML class named **SalariedTaxPayer** that inherits from **TaxPayer**. The class will have one private attribute name **rate**. The rate represents the tax percentage apply to the taxpayer income.
- d. Add a constructor to initialize the class own attributes and its super class.
- e. Implement the class using Java. Use the `super` keyword and concept to implement to the constructor.

- f. Design a class named **TaxCalculator**. The class shall have one method that will process the amount of tax from a tax payer object or its sub classes. Named the method appropriately.
- g. Add another method in the class to display details of a tax payer and the tax amount to be paid. Named the method appropriately.
- h. Implement the **TaxCalculator** class using Java.
- i. Create a class named **TaxCalculatorApp**. This class shall have a `main()` method. The `main()` method shall demonstrate the calculation of tax and display details for tax payers using the information described in Table 3.

Table 3: Tax payer information

Name	IC Number	Income Per Year (RM)
Siti Nurhaliza binti Taruddin	790178-07-6780	10, 679, 568.67
Muhammad Najib bin Abdul Razak	450819-03-4521	568, 344.23
Ananda Krishna	890511-06-2387	80, 231.11

- j. Complete the design with appropriate relationships. Include the **TaxCalculatorApp** class in the design.
- k. All Java codes must be neatly written, comply to the Java coding standard and contains sufficient and descriptive comments.
- l. Compress the UML design, source codes (only .java files) and an output screen shot in one.zip file.
- m. Submit the file in ulearn.

Exercise 7

Some salaried taxpayers receive bonuses. Modify the **SalariedTaxPayer** class to handle this.

- a. Overload the constructor in **SalariedTaxPayer** to accept an additional bonus parameter.
- b. Use the `super` keyword and concept to call the super class constructor.
- c. Update the method that calculates the tax to include bonus in the income.
- d. Update the method that display the tax payer details to include bonus and updated tax.
- e. Create a class named **EnhancedTaxCalculatorApp**. This class shall have a `main()` method. The `main()` method shall demonstrate the calculation of tax and display details for tax payers using the information described in Table 4.

- f. Use `List` and `ArrayList` to loop over the tax payers.

Table 4: Updated tax payer information

Name	IC Number	Income Per Year (RM)	Bonus (RM)
Siti Nurhaliza binti Taruddin	790178-07-6780	10, 679, 568.67	1, 682, 676.67
Muhammad Najib bin Abdul Razak	450819-03-4521	568, 344.23	10% of income per year
Ananda Krishna	890511-06-2387	80, 231.11	15% of monthly income

- g. Update the UML design and include the **EnhancedTaxCalculatorApp** class in the design.
- h. All Java codes must be neatly written, comply to the Java coding standard with sufficient and descriptive comments.
- i. Compress the UML design, source codes (only .java files) and an output screen shot in one.zip file.
- j. Submit the file in ulearn.

Exercise 8

Business owners pay tax at a different rate – which is 15% from annual income.

- a. Using the knowledge comprehended in the previous exercises and lecture, use inheritance to design tax structure for the business owners.
- b. Apply the method overriding concept to calculate and display information about the business owners and tax amount to be paid.
- c. Create another class with a `main()` method to demonstrate the business owners tax calculation using the information in Table 5.

Table 5: Business owners information

Name	IC Number	Income Per Year (RM)
Fahrin bin Ahmad	801010-01-6775	889,331.87
Abdul Fattah bin Amin	901214-15-6531	1, 568, 344.23

- d. Use List and ArrayList to loop over the business owner's information.
- e. Update the UML design and include all classes in this exercise.
- f. All Java codes must be neatly written, comply to the Java coding standard with sufficient and descriptive comments.
- g. Compress the UML design, source codes (only .java files) and an output screen shot in one.zip file.
- h. Submit the file in ulearn.

End of Document

