

Lab 2

Object Oriented Modeling

Learning outcome:

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

1. be able to analyze the problem and construct an object-oriented program.
2. be able to understand the relationship among classes.

Dateline: End of week 3 - depending on your lab hour. Submit to your demonstrator before the lab session ends. Upload to Putrablast.

****** Copy or other forms of cheating is forbidden. The standard penalty for the first offence is to award 0 to all parties concerned.

1. The UML diagrams for the **Course** class is shown as Figure 1 below:

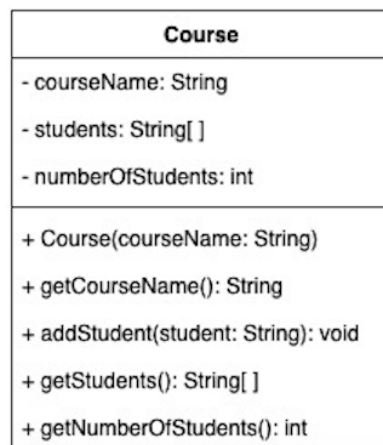


Figure 1

Write a class named **Course** to represent course. A **Course** object can be created using the constructor by passing a course name. Students can be added by using **addStudent(String student)** method and return all the students for that course using the **getStudents()** method. Write a complete test program named **TestCourse** that can add three courses with more than five students for each courses. Display the number of student and students name for that courses as shown below:

```
Number of students in Programming II course : 6
1. Siti Aminah Muhammad
2. Halim Zainal Abidin
3. Jason Lim
4. ..
5. ..
Number of students in Database Systems course : 15
1. Fatimah Ahmad
2. Sarah Goh
..
```

2. Smart Beauty Shop needs a Fat Gram Calculator that calculates the percentage of calories from fat, given a food item, the total of calories and fat grams in a food item. The number of calories from fat cannot be greater than the total of calories in the food item. If the program determines that the number of calories from fat is greater than the number of calories in the food item, it should display an error message indicating that the input is invalid.

Fat contains 9 calories per gram. The calculator should display the percentage of the calories that come from fat. The percentage of calories from fat can be calculated as follows:

Calories from fat = grams of fat * 9

Percentage of calories from fat = (calories from fat / total calories) * 100

If the calories from fat are less than 30% of the total calories of the food, it should also display a message indicating the food is low fat.

The number of calories from fat cannot be greater than the total of calories in the food item. If the program determines that the number of calories from fat is greater than the number of calories in the food item, it should display an error message indicating that the input is invalid.

Construct the UML, class and an application that creates a food object. Your class should have the following:

- Give an appropriate name for your class
- Define appropriate data member and their data type
- Constructor with and without arguments
- Setter method and getter method
- Other appropriate methods.
- Test your application with food items below:

Food Item	Total Calories	Fat (gram)
Crunchy Peanut	539.7	32.22
Muruku	111.0	7.00
Food A	211.0	7.00
Nasi lemak	644.0	83.0

3. **Relationship among classes.** In lab1 you already have class Identification. Now construct a Book class that contain the following members:
- Four private member variables: name (String), author (an instance of the Identification class), price (double), and qty (int).
 - The public getters and setters: getName(), getAuthor(), getPrice(), setPrice, getQty and setQty().
 - A toString() that returns "book-name by author-name (gender) at email with price". You could reuse the Identification's toString() method, which returns "author-name (gender) at email".

Write a complete test program named **TestBook** to display two different book.