Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

[DES103(Y23S2)-LAB05-REVIEW]

Array of Objects and Visibility Modifiers

Learning Objectives

1. To learn the meaning of instance and static variables and methods.
2. To learn about how to create arrays and arrays of objects.
3. To learn about how to use static and instance variables/methods.
4. To learn about how data can be protected so that only programs in certain scopes/locations can access it.
5. To learn three visibility modifiers: private, default, and public to be used to mark the scope of data accessibility.
6. To learn about how to access the encapsulated variables.

****Remark**** A ***pointer finger* (👉)** refers to an explanation between students and their TA.

5.1 Types of Variables and Methods

- *Instance variables/methods* belong to each individual instance (object).
- *Static variables/methods* are used together or shared among instances of the same type.

  ☞To explain your TA, Student SHOULD try to understand the following example:

  ```
  static int money = 10;
  ```

  The keyword `static` is put in front of the variable type.

5.2 Arrays in Java programming

- ***Arrays*** are used to store multiple values in a single variable, instead of declaring separate variables for each value.
- In Java programming, arrays are 0-based indexing like Python and C/C++.

👉 To answer your TA questions, student SHOULD try to understand the following example:

```
int[] myNumbers = {10, 20, 30, 40};  //initialize Integer arrays

String[] myCars = new String[2];     //initialize String arrays
myCars [0] = "Honda";                //assign values


Animal[] myAnimals =  new Animal[2]; //initialize Class arrays
myAnimals [0] = new Animal("Tiger", “4 legs”); //assign values
```

Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

5.3 The use of static variables/methods

- *Static variables and methods* can be used from both instance or static methods in the class. As they are class's mutual variables and methods, they can also be accessed through the class name.
- *Instance variables and methods* can be used only from instance methods, not from a static method.

5.4 Visibility Modifier

- *Visibility modifier* is a word that is put in front of the class, properties, constructor, or methods to indicate the scope of visibility.
- Java programming organizes files into groups according to their functionality using ***Package***.

Visibility /Modifier	**Scope**	**UML Notation**
`public`	classes, methods, and data fields are accessible from any class on any package	**+**
`Protected`	classes, methods, and data fields are accessible from any class on the same package or outside of the package for its subclass	**#**
`Default(nothing)`	classes, methods, and data fields are accessible only from within the same package	
`private`	classes, methods, and data fields are accessible only from within its own class	**–**

👉 To answer your TA questions, student SHOULD try to understand the access level table:

Modifier	**Class**	**Package**	**Subclass**	**Global**
`public`	✅	✅	✅	✅
`protected`	✅	✅	✅	❌
`Default (nothing)`	✅	✅	❌	❌
`private`	✅	❌	❌	❌

5.5 Encapsulation

- The concept of protecting a variable from being directly modified is called ***encapsulation***.
- When the programmer would like to allow the users to access the encapsulated variables, he/she opens a new channel with higher visibility to allow the users to use it.

👉 To answer your TA questions, student SHOULD try to understand the following example:

```
Classroom----------------------------------------------------
package Classroom;
public class Student {
        private String name;                    //private data member
        public String getName() {               //getter method for name
                return name;
        }
        public void setName(String name){       //setter method for name
                this.name=name
        }
}

RegistrationDepartment-------------------------------
package RegistrationDepartment;
 public class RegistrationDepartment_Testing {
     public static void main(String[] args){
                Student student01 = new Student();          //creating instance
                student01.setName("Adam");                   //setting value
            System.out.println(student01.getName());        //getting value
     }
 }
```

Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

[DES103(Y23S2)-LAB05-EXERCISES]

Students MUST adhere to the lab instructions and regulations provided below.
Please consult your TA to review your completed exercises and submit them on Google Classroom.

Be noticed that for all lab exercises, you need to define your *Java* project as the following name format:

<StudentID>_<Lab number>_<Exercise name>

If your student's ID is 6722300208, the name format of your java project should be:
"6422300208_LAB05_FamilyMember" for exercise 1, 2 and 3.
"6422300208_LAB05_BankAccount" for exercise 4 and 5.

Exercise 1: (2 points)

Project Name: <Student_ID>_LAB05_FamilyMember

Instruction : Download *FamilyMember.java* from Google Classroom and complete and a `FamilyMember` class in the following UML.

`FamilyMember`	Description

<u>`familyName="Hilton": String`</u>
`firstName: String`
<u>`commonFund=100000.00: double`</u>
`privateFund: double`
	The family name
	The first name of the family member
	The amount of common fund of the family
	The amount of private fund of the family member

`FamilyMember(firstName: String, privatefund: double)`
 	
`printPrivateFund(): void`
	A 2-arguments constructor that assigns the input `firstname` and privateFund to its corresponding properties.
	
	Print out the privateFund of the instance of FamilyMember in the following format:
	`firstname familyName has $privateFund`

Exercise 2: (2 points)

Project Name: <Student_ID>_LAB05_FamilyMember

Instruction : Download *FamilyMemberTesting.java* from Google Classroom and complete a class `FamilyMemberTesting` in the following tasks.

a) Use the dot operator in the `FamilyMemberTesting` class, and print out `familyName` with `commonFund` in the following format:

```
The familyName family has $commonFund
```

Your running output should be as below:

```
The Hilton family has $100000.0
```

b) Create an `arrayFamily` array to add four members of the `Hilton` family. Each member is assigned as an object of the `FamilyMember` class, and their argument values are as below:

Order	**firstName**	**familyName**	**privateFund**
1	John	Hilton	254639.12
2	Erika	Hilton	187346.56
3	James	Hilton	56783.12
4	Paris	Hilton	12124.88

c) Use the dot operator to call the `printPrivateFund` method for printing all family members. Your running output should be as below:

```
John Hilton has $254639.12
Erika Hilton has $187346.56
James Hilton has $56783.12
Paris Hilton has $12124.88
```

Exercise 3: (2 points)

Project Name : <Student_ID>_LAB05_FamilyMember

Instruction : Update the `FamilyMember` class in the following tasks.

a) Update the `FamilyMember` class that you created in Exercise 1. Add the two additional methods as below.

1) void `contributeToCommonFund(double amount)` which <u>transfers</u> the input amount from the instance's `privateFund` to the `commonFund`

2) *static* void `payFromCommonFund(double amount)` which <u>deducts</u> the input amount from the `CommonFund`.

b) Update the `FamilyMemberTesting` class that you created in Exercise 2, and do in following:

1) The two members in the `arrayFamily` array use the dot operator to call the `contributeToCommonFund` method that `Erika Hilton` and `Paris Hilton` can contribute their `privateFund` $10000 to the `commonFund` of the family.

2) All members in the `arrayFamily` array use the dot operator to call the `printPrivateFund` method for printing their `PrivateFund`.

```
John Hilton has $254639.12
Erika Hilton has $177346.56
James Hilton has $56783.12
Paris Hilton has $2124.879999999999
```

c) Update the `FamilyMember` class that you created in Exercise 1.

Add one additional method as below.

1) *static* void printFamilyFund() which prints the input amount from the `CommonFund` of the family.

d) Use the dot operator to call the `printFamilyFund` method for printing the `CommonFund` of the family in the following format:

```
The familyName family has $commonFund
```

Your running output should be as below:

```
The Hilton family has $120000.0
```

From the exercise 1, 2 and 3, the running output of **<Student_ID>_LAB05_FamilyMember** should be as below:

```
# COMMON FUND OF THE FAMILY
The Hilton family has $100000.0
--------------------------

# PRIVATE FUND
John Hilton has $254639.12
Erika Hilton has $187346.56
James Hilton has $56783.12
Paris Hilton has $12124.88
--------------------------

# CONTRIBUTION OF PRIVATE FUND
John Hilton has $254639.12
Erika Hilton has $177346.56
James Hilton has $56783.12
Paris Hilton has $2124.879999999999
--------------------------

# UPDATED COMMON FUND OF THE FAMILY
The Hilton family has $120000.0
--------------------------
```

Exercise 4: (2 points)

Project Name : <Student_ID>_LAB05_BankAccount

Instruction : Download *BankAccount.java* from Google Classroom and complete a *BankAccount* class in the following tasks.

a) Create a `Developer` package and put *BankAccount.java* in this package. Put appropriate keywords and visibility modifiers (`static`, `private`, and `public`) in the program below so that it fulfills the following requirements.

1) `numAccount` keeps the number of `BankAccount` objects that has been created. The `numAccount` must be accessible anywhere.
2) `ownerName`, and `balance` variables can only be accessed inside of the `BankAccount` class, but not anywhere else.
3) `accountNumber` can only be accessed inside of the package `Developer`, but not anywhere else.
4) `deposit` methods must be accessible anywhere, and `withdraw` methods can only be accessed from any class on the same package or outside of the package for its subclass
5) `printInfo` method can only be accessed inside of the `BankAccount` class and within the package `Developer`, but not anywhere else.
6) `BankAccount` constructor must be accessible anywhere and any class on the same package or outside of the package for its subclass

```
package Developer;
public class BankAccount {
        //______ String ownerName;
        //______ accountNumber;
        //______ double balance;
        //______ int numAccount;

        //______ BankAccount(String ownerName, String accountNumber, double balance) {
                this.ownerName = ownerName;
                this.accountNumber = accountNumber;
                this.balance = balance;
                numAccount++;
        }

        //______ void deposit(double amount) {
                balance = balance + amount;
                System.out.println("Deposit: $"+amount);
        }

        //______ void withdraw(double amount) {
                if (balance > amount) {
                        balance = balance - amount;
                        System.out.println("Withdraw: $"+amount);
                }else {
                        System.out.println("Not enough balance!");
                }
        }
}
```

Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

b) Download TestBankAccount1.java from Google Classroom and put it in the package Developer.

```
package Developer;
public class TestAccount1 {
public static void main(String [] args){
    BankAccount acc = new BankAccount("Paris Hilton","127-983-3847", 1000000.00 );
          System.out.println(acc.balance);
          }
}
```

You must not be able to run this file if you do problem 4 b) correctly.

👉**TA Q&A:** Do you know why? Explain the reason to your TA to get a pass on this problem.

c) Download TestBankAccount2.java from Google Classroom and put it in another package called `Outside`.

```
package Outside;
import Developer.BankAccount;

public class TestBankAccount2 {
public static void main(String [] args){
   BankAccount acc = new BankAccount("Paris Hilton","127-983-3847",
  1000000.00 );
  System.out.println(acc.accountNumber);
 }
}
```

You must not be able to run this file if you do problem 4 c) correctly.

👉**TA Q&A:** Do you know why? Explain the reason to your TA to get a pass on this problem.

Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

Exercise 5: (2 points)

Project Name : <Student_ID>_LAB05_BankAccount

Instruction : Download *BankAccount01_Testing.java* and *BankAccount02_Testing.java* from Google Classroom and complete the following tasks.

a) Put *BankAccount.java* in the `Developer` package, and answer in the following questions: Exercise 5-a-1 and Exercise 5-a-2.

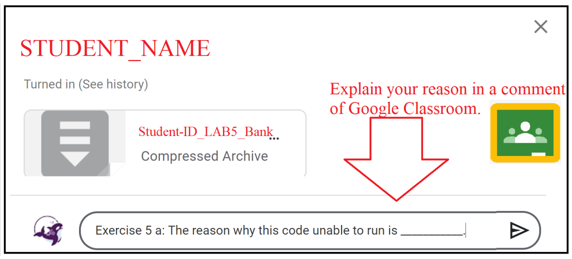
```
package Developer;
public class BankAccount1_Testing {
        public static void main(String[] args) {

                BankAccount account1 = new BankAccount
                                        ("Paris Hilton", "127-983-3847", 1000000.00);

                //System.out.println("The balance of account1 is $" + account1.balance);

                //deposit $300 for account1
                //print out the balance of account1
        }
}
```

Exercise 5-a-1

```
//System.out.println("The balance of account1 is $" + account1.balance);
```

Uncomment the above code and run this Java project. If students put appropriate keywords and visibility modifiers in the Exercise 4-a correctly, you MUST NOT be able to run this code line in this Exercise 5-a-1.

👉 **TA Q&A:** Do you know why? Explain your answer **Exercise 5-a-1** in a comment of the Google Classroom as shown at below figure:


STUDENT_NAME
Turned in (See history)
Student-ID_LAB5_Bank...
Compressed Archive
Explain your reason in a comment of Google Classroom.
Exercise 5 a: The reason why this code unable to run is ________

Exercise 5-a-2

```
//deposit $300 for account1
```

Use the dot operator to call the `deposit` method to add $300 into the `account1`.

```
//print out the balance of account1
```

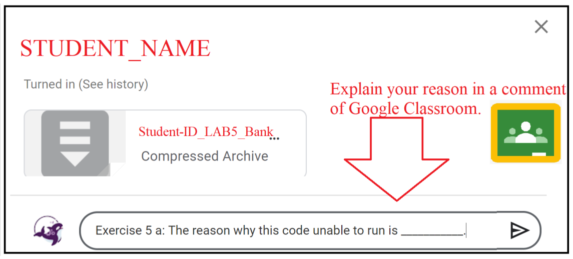
Apply the Encapsulation concept to solve the problem in Exercise 4-b-1 and print out the balance of `account1`.

Your running output should be as below:

```
Deposit: $300.0
The balance of account1 is $1000300.0
```

Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

👉 **TA Q&A:** Do you know why? Explain your answer **Exercise 5-a-2** in a comment of the Google Classroom as shown at below figure:


STUDENT_NAME
Turned in (See history)
Student-ID_LAB5_Bank...
Compressed Archive
Explain your reason in a comment of Google Classroom.
Exercise 5 a: The reason why this code unable to run is ________

c) Put *BankAccount02_Testing.java* in the `Outside` package, and complete the answer in the following question: Exercise 5-b-1 and Exercise 5-b-2.

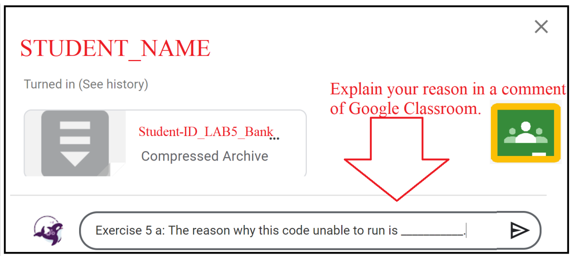
```
package Outside;
import Developer.BankAccount;
public class BankAccount2_Testing {
        public static void main(String [] args){
                BankAccount account2 = new BankAccount
                                ("Paris Hilton","127-983-3847", 1000000.00);

                //account2.printInfo();
                //account2.withdraw(300.00);
                //System.out.println("The balance of account1 is $" +
account2.balance);
    }
}
```

Exercise 5-b-1

`//account2.printInfo();`

Uncomment the above code and run this Java project. If students put appropriate keywords and visibility modifiers in the Exercise 4-a correctly, you MUST NOT be able to run this code line in this Exercise 5-b-1.

👉 **TA Q&A:** Do you know why? Explain your answer **Exercise 5-b-1** in a comment of the Google Classroom as shown at below figure:


STUDENT_NAME
Turned in (See history)
Student-ID_LAB5_Bank...
Compressed Archive
Explain your reason in a comment of Google Classroom.
Exercise 5 a: The reason why this code unable to run is ________

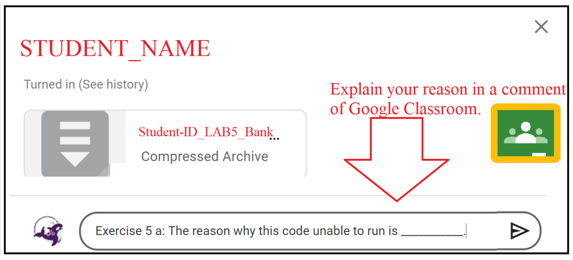
Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

Exercise 5-b-2

```
//account2.withdraw(300.00);
```

Uncomment the above code and run this Java project. If students put appropriate keywords and visibility modifiers in the Exercise 4-a correctly, you MUST NOT be able to run this code line in this Exercise 5-b-2.

👉 **TA Q&A:** Do you know why? Explain your answer **Exercise 5-b-2** in a comment of the Google Classroom as shown at below figure:


STUDENT_NAME
Turned in (See history)
Student-ID_LAB5_Bank...
Compressed Archive
Explain your reason in a comment of Google Classroom.
Exercise 5 a: The reason why this code unable to run is ________

Exercise 5-b-3

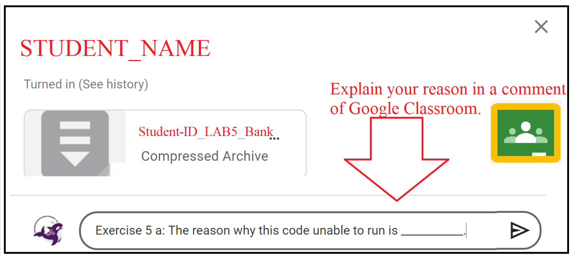
```
//System.out.println("The balance of account1 is $" + account2.balance);
```

Keep the comment in `//account2.withdraw(300.00)` and apply the Encapsulation concept to solve the problem in Exercise 5-b-3 and print out the balance of `account2`.

Your running output should be as below:

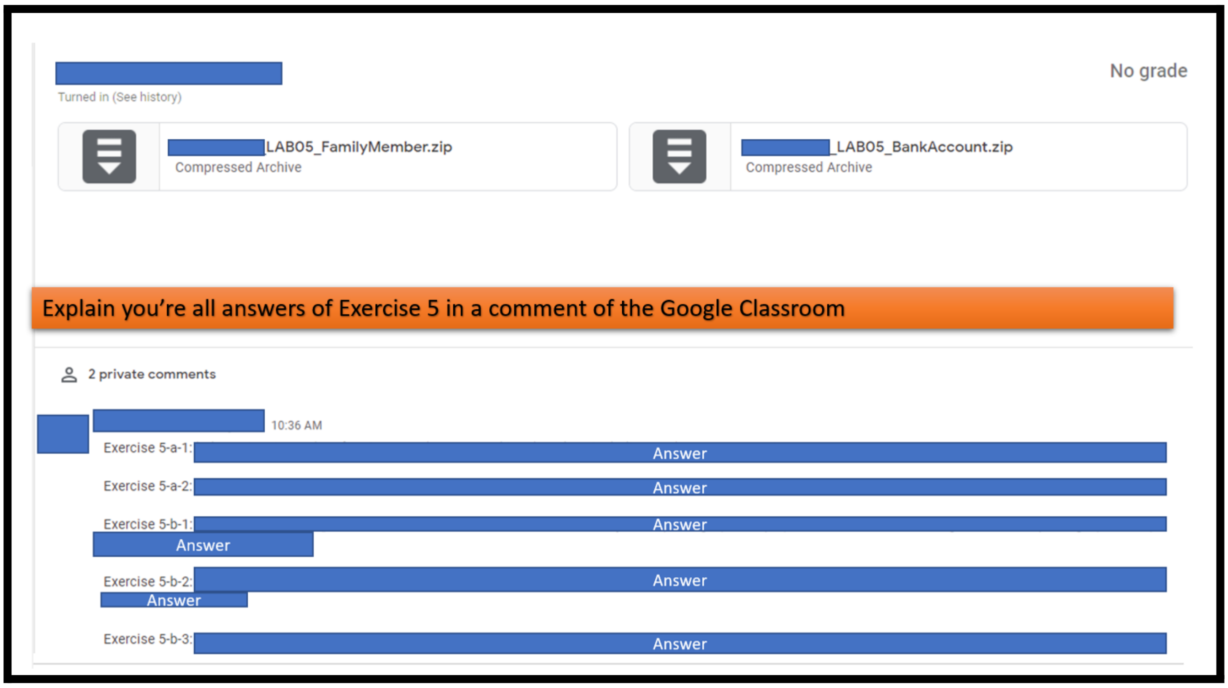
```
The balance of account2 is $1000000.0
```

👉 **TA Q&A:** Do you know why? Explain your answer **Exercise 5-b-3** in a comment of the Google Classroom as shown at below figure:


STUDENT_NAME
Turned in (See history)
Student-ID_LAB5_Bank...
Compressed Archive
Explain your reason in a comment of Google Classroom.
Exercise 5 a: The reason why this code unable to run is ________

Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

👉**TA Q&A:** Explain you're all answers of Exercise 5 in a comment of the Google Classroom



[redacted]
Turned in (See history)

No grade

[redacted]LAB05_FamilyMember.zip
Compressed Archive

[redacted]LAB05_BankAccount.zip
Compressed Archive

Explain you're all answers of Exercise 5 in a comment of the Google Classroom

2 private comments

[redacted] [redacted] 10:36 AM

Exercise 5-a-1: Answer

Exercise 5-a-2: Answer

Exercise 5-b-1: Answer
Answer

Exercise 5-b-2: Answer
Answer

Exercise 5-b-3: Answer