*Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, and Dr. Akkarawoot Takhom* {sasiporn.us, akkharawoot.aj, jessada.aj}@siit.tu.ac.th

# DES103 LAB01:

## `Class` Component, Basic `printout` statement, The `dot` operator, and The `new` operator

## Learning Objectives

- To learn how to create a Java project by Eclipse IDE for writing and running Java codes
  **An integrated development environment (IDE)*
- To learn how to declare variables and functions in Java programming
- To learn how to write a `main` method before get printed out at the console
- To learn about the `dot operator (.)`
- To learn about the `+ operator` for concatenate String and a number
- To learn about basic `print`out statements

## 1.1 What is OOP?

Java is an *object-oriented programming (OOP)* language. This means that everything in Java, except of the primitive types (int, float, double, and etc.), is an object. However, what is an object? The concept of using classes and objects is to encapsulate state and behavior into a single programming unit. Java objects are similar to real-world objects.
For example, we can create a car object in Java, which will have properties like current speed and color, and behavior like: accelerate and park.

## 1.2 Creating a Java Project

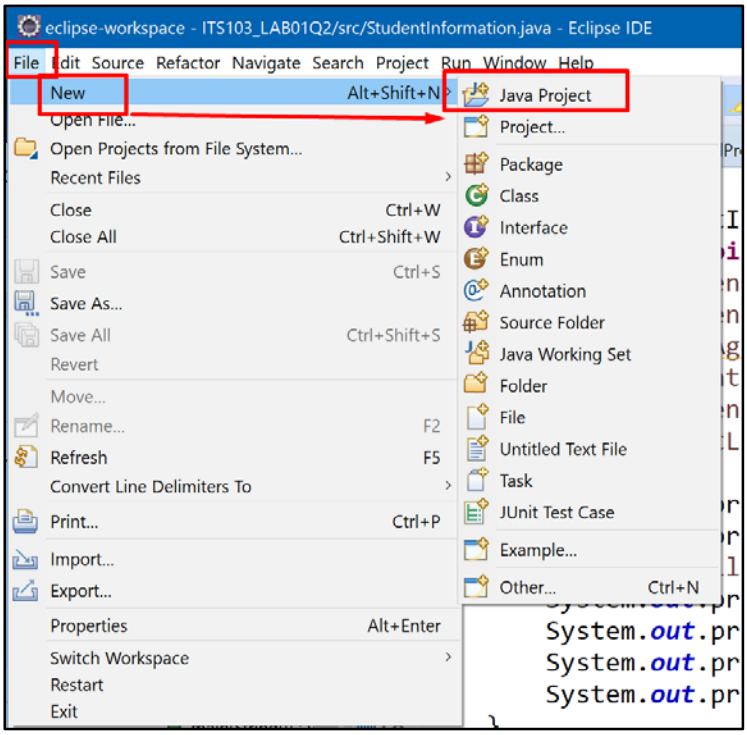
Java classes are the blueprints of which objects are created. Let's create a class that print out String "Hello World".

**Print Statement:**

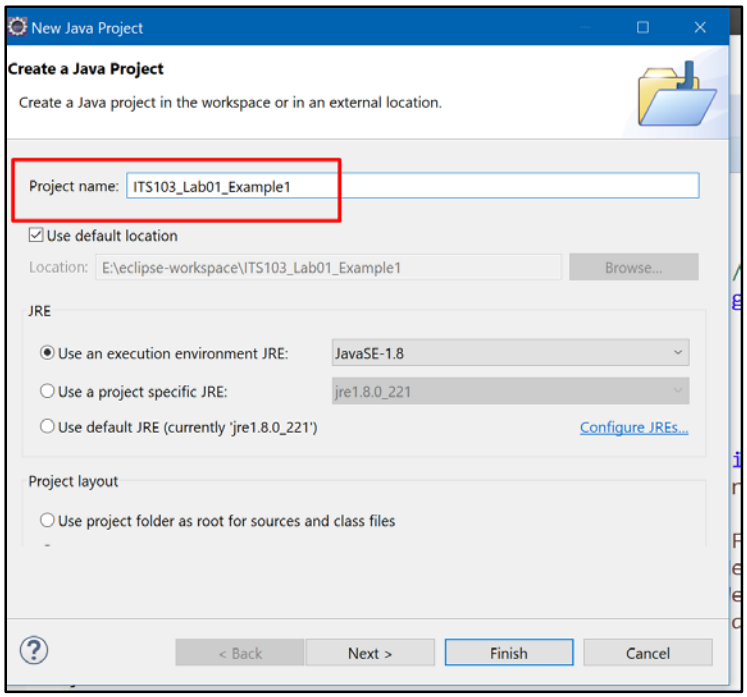
- `System.out.print (…)` prints the argument
- `System.out.println(…)` prints the argument then add a new line after finish printing
- `System.out.printf("… %f ",val);` print input argument with format
  Ex. %f means float and %d means integer

**Step1**: Create a *Java Project*

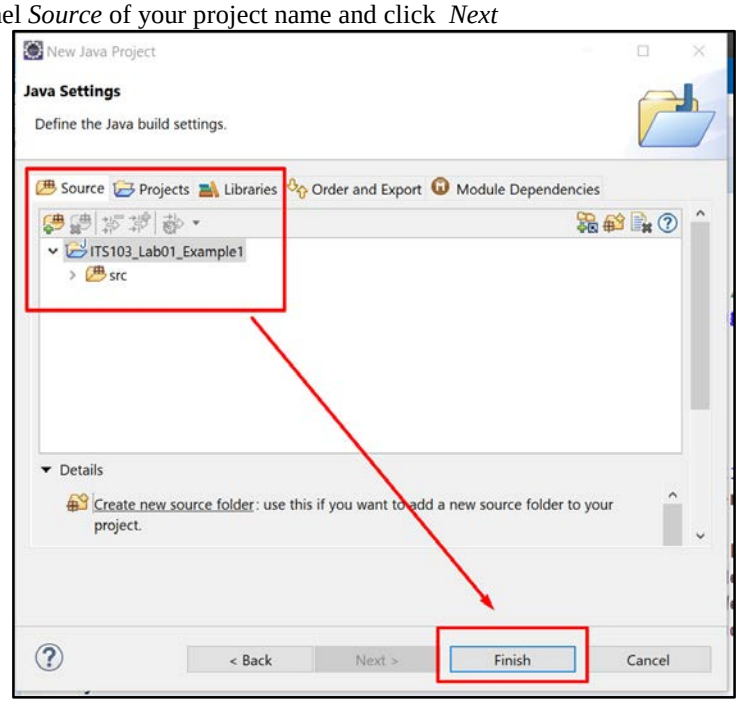
File → New → Java Project




**Step2**: Define your *Project name* and click *Next*

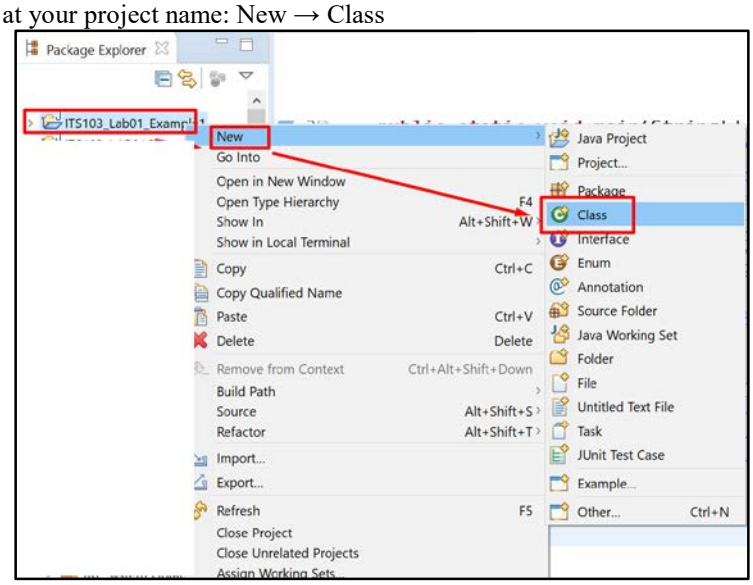


**Step3**: Check the panel *Source* of your project name and click *Next*

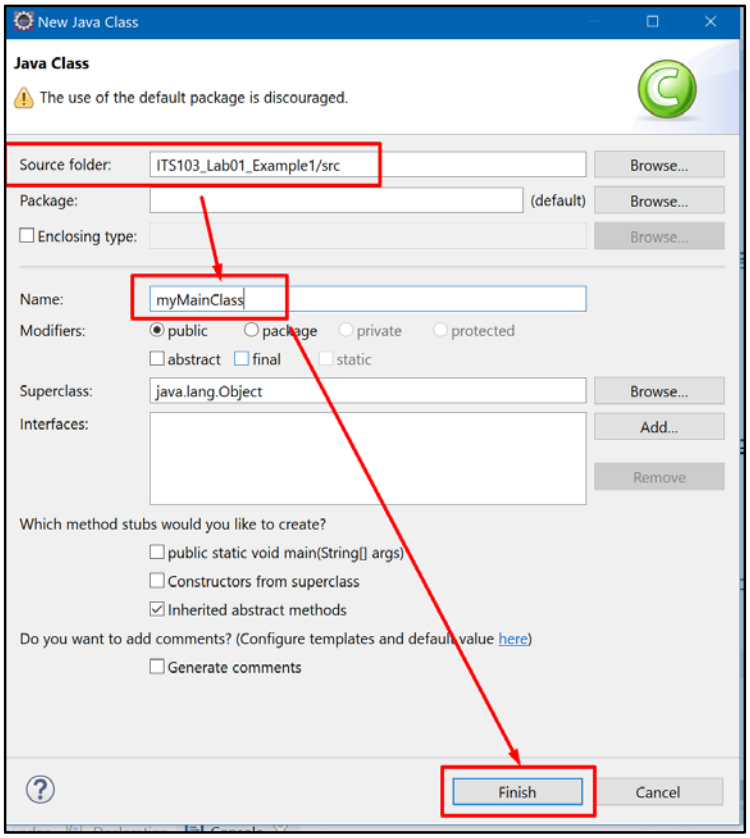



**Step4**: Define your *Class*

Right-click at your project name: New → Class

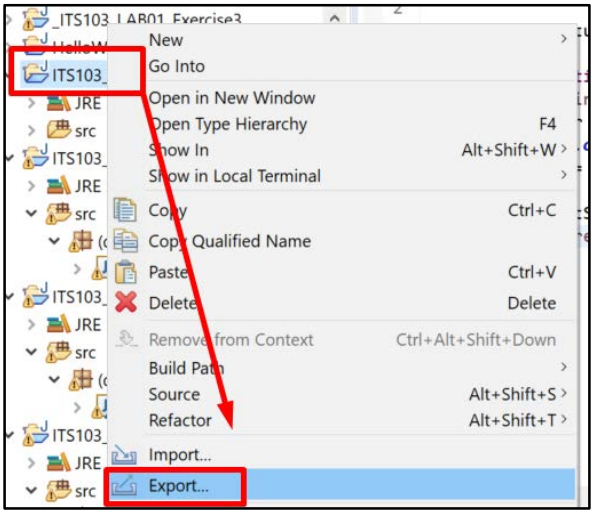


**Step5**: Check your *Source* folder,
Define your class *Name* and click *Finish*

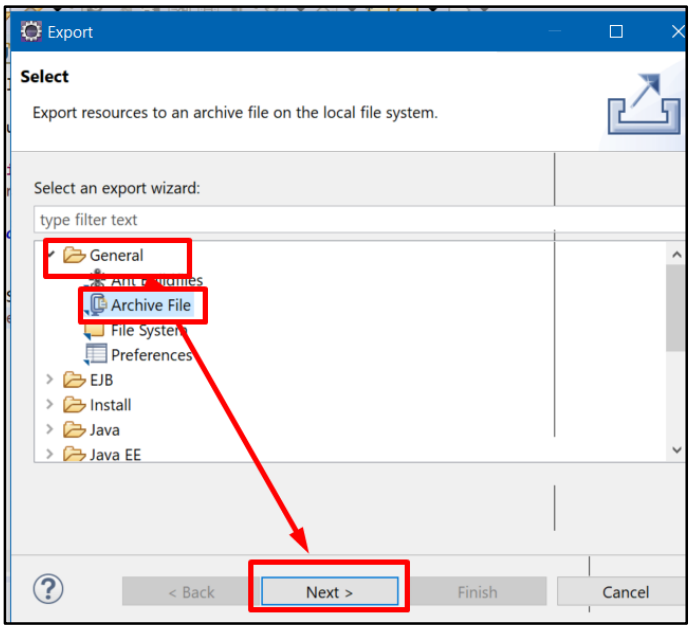



## 1.3 Exporting a Java Project

**Step1**: Right click at your finished java project and select at Export...

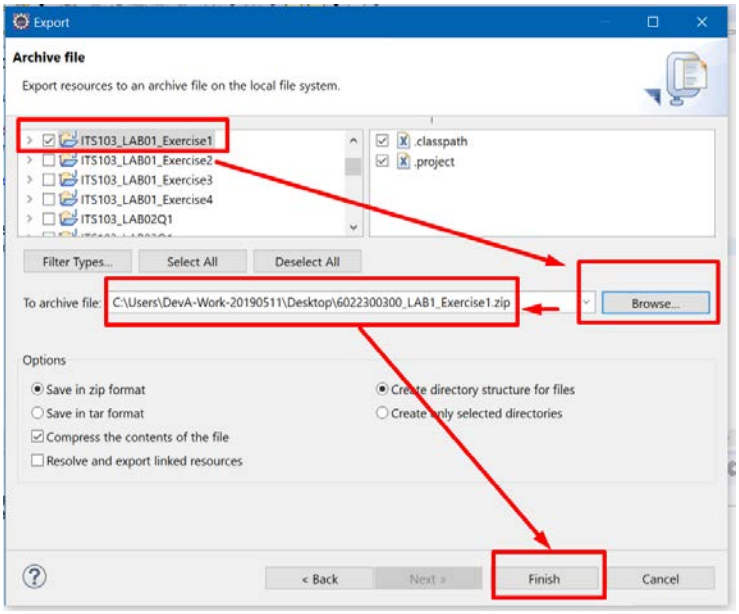


**Step2**: Select General → Archive File → Next >




**Step3**:

- Select your finished java project → Browse…
- Select your file location
  (e.g., Desktop)
- Define name in the following name format:
  <StudentID>_<Lab number>_<Exercise number>
  (e.g., 6022300300_LAB1_Exercise1.zip)
- Click Finish and check your file on Desktop



*Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, and Dr. Akkarawoot Takhom* {sasiporn.us, akkharawoot.aj, jessada.aj}@siit.tu.ac.th

# LAB01 Exercises

## Students should follow lab instructions and regulations.
## Ask your TA to check your finished exercises and attach them to Google Class.

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

<Student ID>_<Lab number>_<Exercise number>

If your student's ID is 6122300300, the name format of your java project should be:

6422300208_LAB01_Example

## Exercise 1 *(2 points)*

**Project Name**: <Student ID>_LAB01_Rectangle

Write a JAVA class, called `Rectangle`, that has two properties: `width` and `length`. This class has two `constructors`. The first constructor takes no argument, and the `constructor` sets the width to 1 and sets the length to 1. The second constructor takes two arguments that set those two properties. This class has five methods, as follows.

1. `double findArea()`: it computes and returns the area of the rectangle.
2. `double findPerimeter()`: it computes and returns the perimeter of the rectangle.
3. `double findDiagonal()`: it computes and returns the diagonal of the rectangle.
4. `boolean isSquare()`: it returns true if the rectangle is a square; otherwise, false.
5. `void printBasicInfo()`: it prints the following two lines.
    a. The width is [`width`].
    b. The length is [`length`].

Note: [`property`] means the value of the property. For example, [`width`] means the value of the property named width.

## Exercise 2 (2 points)

**Project Name**: <Student ID>_LAB01_Rectangle

Write a JAVA class, called `TestRectangle`, that tests the `Rectangle`. It has only the `main` method. In the `main` method, do the following.

1. Use the keyword new to create an object of `Rectangle` with the no-argument `constructor` and name this object `box1`.
2. Print the basic information of `box1`.
3. Print the perimeter of `box1`.
4. Print the diagonal of `box1`.
5. If box1 is a square box, print "`It is a square box.`" Otherwise, print "`It is not a square box.`"
6. Repeat 1-5 with another object that is created with another `constructor`. You may name this object box2.

## Exercise 3 *(2 points)*

**Project Name**: <Student ID>_LAB01_BankAccount

Write a JAVA class, called `Person`, that has five properties: `name, surname, sex, occupation,` and `organization`. This class has only one `constructor` that takes five arguments for setting all parameters. It has only one method called `printInfo()`, and this method prints the following five lines.

- `Name: [name]`
- `Surname: [surname]`
- `Sex: [sex]`
- `Occupation: [occupation]`
- `Organization: [organization]`

## Exercise 4 *(2 points)*

**Project Name**: <Student ID>_LAB01_BankAccount

Write a JAVA class, called `BankAccount`, that has three properties: `person, accountNumber,` and `balance`. The property `person` is an object of the class `Person`. This class has one constructor that takes seven arguments: `name, surname, sex, occupation, organization, accountNumber,` and `balance`. The first five arguments are used to set the property person by creating a new object of `Person` (with the constructor in Problem 3). The other two arguments are used to set the properties `accountNumber` and balance, respectively. This class has five methods, as follows.

1. `void deposit(double x)`: it updates the balance with respect to the new deposit x.
2. `void withdraw(double x)`: it updates the balance with respect to the new withdrawal.
3. `void printInfo()`: it prints the following seven lines.
    - `Name: [name]`
    - `Surname: [surname]`
    - `Sex: [sex]`
    - `Occupation: [occupation]`
    - `Organization: [organization]`
    - `Account Number: [accountNumber]`
    - `Balance: [balance]`
4. `void printBalance()`: it prints "`Balance = [balance] million USD`"
5. `double convertBalanceToTHB()`: it converts the balance from USD to THB and returns the amount in THB.

## Exercise 5 *(2 points)*

**Project Name**: <Student ID>_LAB01_BankAccount

Write a JAVA class, called `TestBankAccount`, that test the `Person` and the `BankAccount`. It has only the main method. In the main method, do the following.

1. Create an object of `BankAcount` with the following pieces of information: `name = Wang, surname = TaLu, sex = Male, occupation = Actor, organization = SIIT, accountNumber = 000-000-0000,` and `balance = 10`.
2. Print information.
3. Change the name, surname, and sex to yours.
4. Print information.
5. Call the deposit method of the object you created in step 1 to deposit 15 million USD to the account.
6. Print an updated balance.
7. Call the withdraw method to withdraw 5 million USD from the account.
8. Print an updated balance.
9. Print the balance in THB.