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)-LAB01-rev01]

`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 printout statements

1.1 What is OOP?

- **Programming Paradigm in Java**: Java exemplifies an *object-oriented programming (OOP)* language, categorizing nearly all elements, excluding primitive types (e.g., int, float, double), as objects.
- **Conceptual Alignment of Objects in Java**:
  Objects in Java are conceptually aligned with tangible real-world entities, underscoring the language's OOP paradigm.
- **Application of Object-Oriented Design Principles:**
  A practical example involves creating a *`"car"`* object in Java, endowed with properties such as current speed and color, and featuring behaviors like acceleration and parking.
- **Representation of Complex Systems in Java:** This instantiation of classes and objects illustrates object-oriented design principles, facilitating the representation of complex systems akin to real-world scenarios.

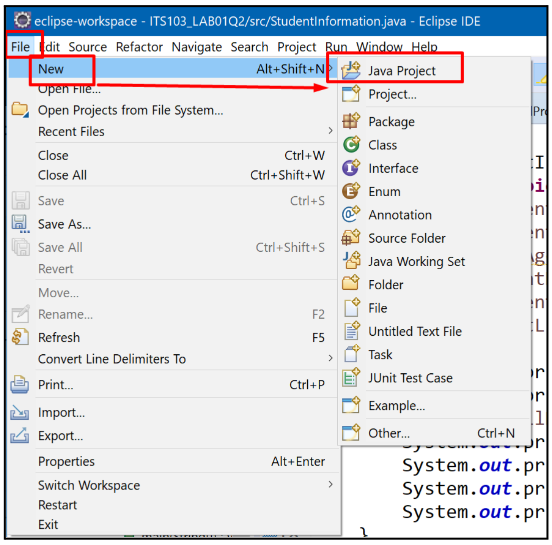
Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom
{sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

1.2 Creating a Java Project

Java classes serve as the foundational blueprints from which objects are instantiated.
To exemplify this concept, let us generate a class designed to output the string *"Hello World."*
When engaging with the Java print statements, it's imperative to comprehend their distinct functionalities:

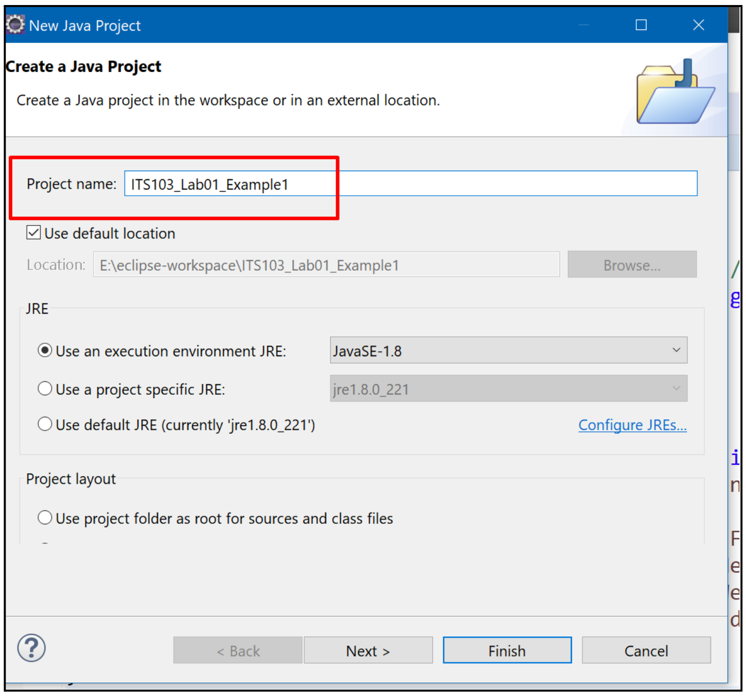
- `System.out.print(…)`: This statement prints the specified argument.
- `System.out.println(…)`: The `println` statement not only prints the provided argument but also appends a new line upon completion of the printing process.
- `System.out.printf("… %f ", val)`:
  This statement allows for the printing of input arguments with a specified format.
  - For instance, the format specifier *%f* denotes a float, and *%d* corresponds to an integer. Understanding these format specifiers is crucial for precision in output formatting.

Step1: Create a *Java Project :*
`File → New → Java Project`



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.

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



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: ITS103_Lab01_Example1

☑ Use default location

Location: E:\eclipse-workspace\ITS103_Lab01_Example1

JRE

● Use an execution environment JRE: JavaSE-1.8

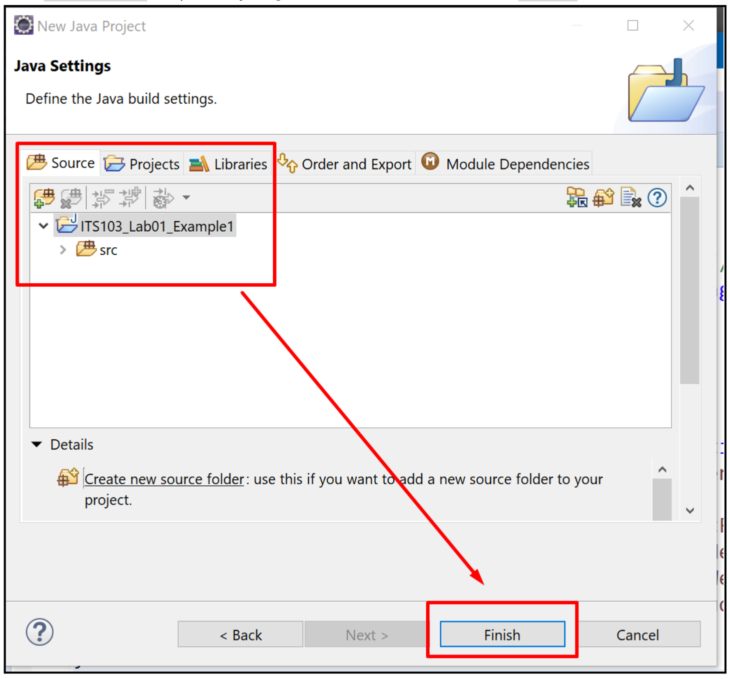
○ Use a project specific JRE: jre1.8.0_221

○ Use default JRE (currently 'jre1.8.0_221')

Project layout

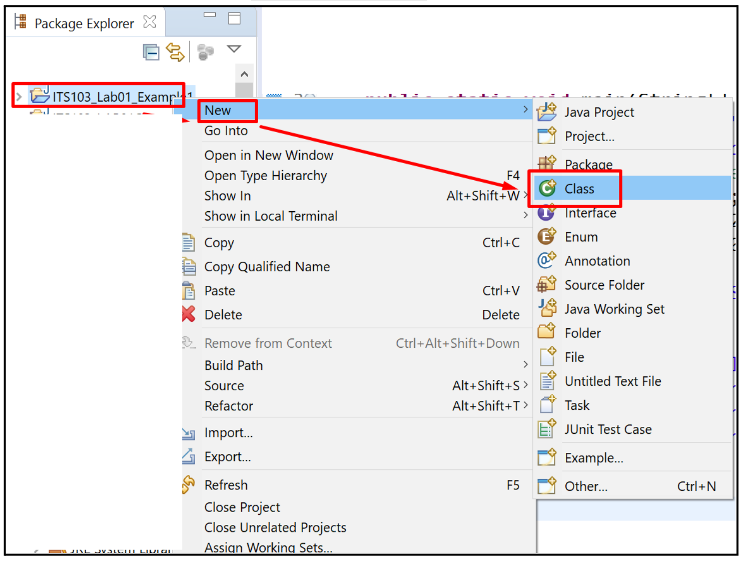
○ Use project folder as root for sources and class files

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



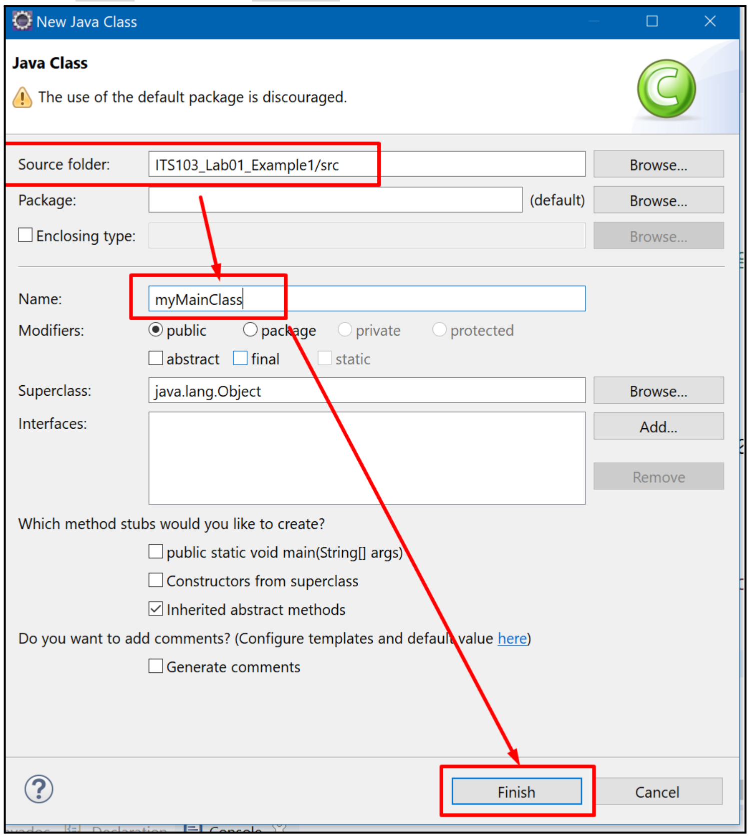
Step4: Define your *Class*

Right-click at your project name: `New → Class`



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.

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


New Java Class

Java Class

The use of the default package is discouraged.

Source folder: ITS103_Lab01_Example1/src

Package: (default)

☐ Enclosing type:

Name: myMainClass

Modifiers: [✓ public] package private protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☑ Inherited abstract methods

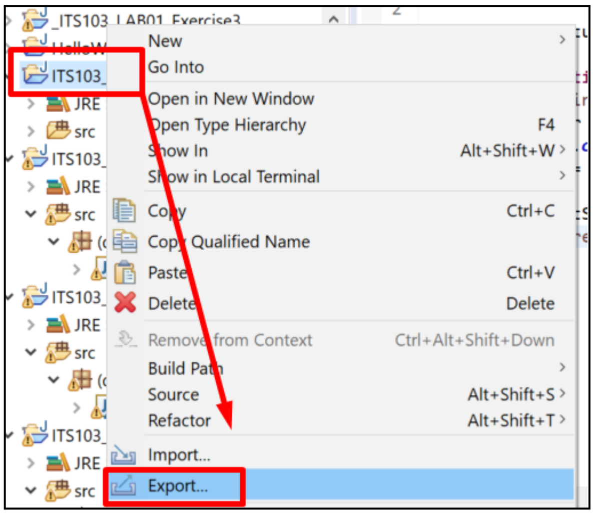
Do you want to add comments? (Configure templates and default value here)

☐ Generate comments

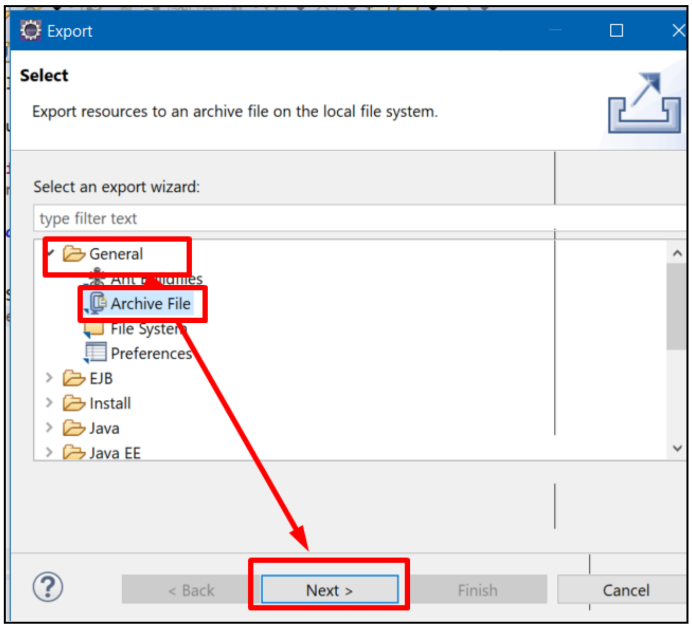
Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom
{sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

1.3 Exporting a Java Project

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



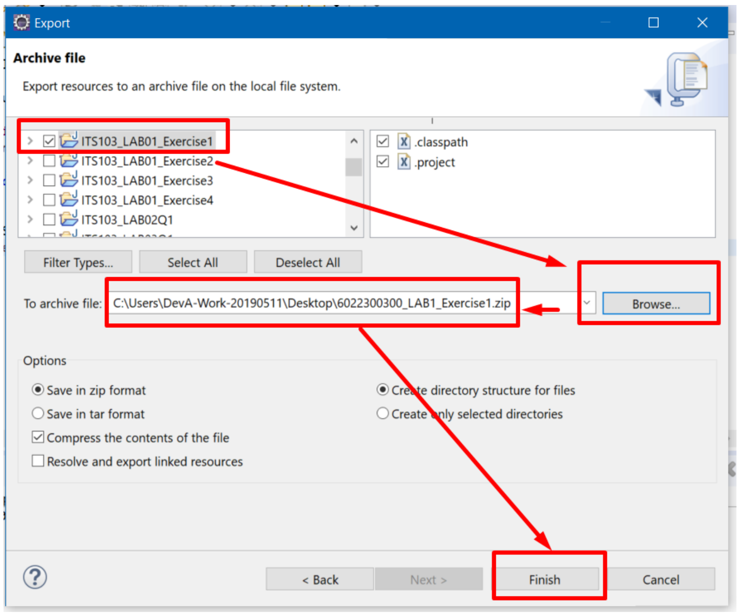
Step2: Select `General → Archive File → Next →`



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.

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



Export

Archive file

Export resources to an archive file on the local file system.

- [✓] ITS103_LAB01_Exercise1
- [ ] ITS103_LAB01_Exercise2
- [ ] ITS103_LAB01_Exercise3
- [ ] ITS103_LAB01_Exercise4
- [ ] ITS103_LAB02Q1

- [✓] .classpath
- [✓] .project

To archive file: C:\Users\DevA-Work-20190511\Desktop\6022300300_LAB1_Exercise1.zip

Options

- (●) Save in zip format
- ( ) Save in tar format
- [✓] Compress the contents of the file
- [ ] Resolve and export linked resources
- (●) Create directory structure for files
- ( ) Create only selected directories

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)-LAB01-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 Class.

Please note that for all lab exercises, you are required to define *your Java project* using the following naming format:

`<StudentID><LabNumber><ExerciseNumber>`

For instance, if your student ID is 6122300300, the name format of your Java project should be:

6422300208_LAB01_Example

Exercise 1 *(2 points)*

Project Name: <StudentID>_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

***Hint**: Instantiation and Dot Operator*

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/instantiate 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/instantiated with a two-argument `constructor`. You may name this object *box2*.

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 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.

```
1. Name: [name]
2. Surname: [surname]
3. Sex: [sex]
4. Occupation: [occupation]
5. Organization: [organization]
```

Exercise 4 *(2 points)*

Project Name: <Student ID>_LAB01_BankAccount

Write a JAVA class, called `BankAccount`, that has tree 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.

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

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>_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.