# Lab 01 Introduction to Java and OO-basics Review

Q1-2 A Java class - Day
Q3 Java programming in console mode (**`SWING`**, **`AWT`**)
Q4 [Homework] Examine an OO sample program: **`Library`**
Q5 [Homework] calculate the *previous day*
Q6 [Homework] OO Programming from C++ to Java

**[Tasks Q1-3 originally drafted by Mr. HO Wai Kit (BSC2 - 2014)]**

## Preparation Task (Also given in Canvas Announcement before Week 1)

We will write Java programs using **Visual Studio Code** (VS Code) with **Java SE-11** in this course.

You have been told to go through the following steps before this lesson.

> If you haven't taken these steps to setup VS Code, you may do it after this lesson.
> During this lesson, you may also use **repl.it** online instead of VS Code:
> https://repl.it/languages/java
> *"How?"* : Go to this website, setup your account, drag java files to the Files area, then click "Run".

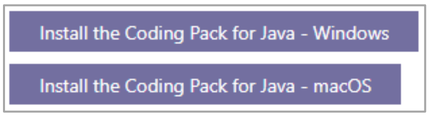
### Step 1. Install **Java SE JDK 11**

Java SE JDK 11 is available for various computer platforms. You can download the right one via https://www.oracle.com/java/technologies/javase-jdk11-downloads.html.
[Alternative: Google: *download java 11*].

When download starts, you will be **required to sign in with (or sign up for) an Oracle account**.

### Step 2. Install **Visual Studio Code**

The installer of *VS Code for Java Developers* is available on Windows and macOS. You can download and install the right one via https://code.visualstudio.com/docs/languages/java.
[Alternative: Google: *visual studio code for java*].


Install the Coding Pack for Java - Windows
Install the Coding Pack for Java - macOS

### Step 3. Set up a master folder for CS2312 programs in your computer.

Create the folder `CS2312_Programs` in your hard disk, e.g. `C:\CS2312_Programs\`
During the semester you will create subfolders for sample programs and exercises within this path.

### Step 4. Create and run the sample Java Program.

Then follow the steps below to create and run the sample java program.

4.1 From `Canvas => CS2312 => Announcements => Preparation for Lab 01`, browse the given code for the sample program. You will need the code in the coming steps.

4.2 Create the program folder `C:\CS2312_Programs\`**`Lab01_Q01`**

4.3 Open the program folder in VS Code:
`VS Code => File => Open Folder => select the folder`
Then create two empty source files named `Main.java` and `Day.java` (Figure 1).

If you get lost, `the Show Explorer button` can help you. (Figure 2)

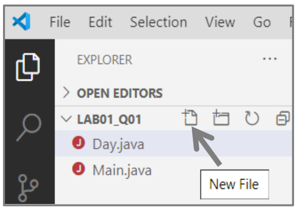
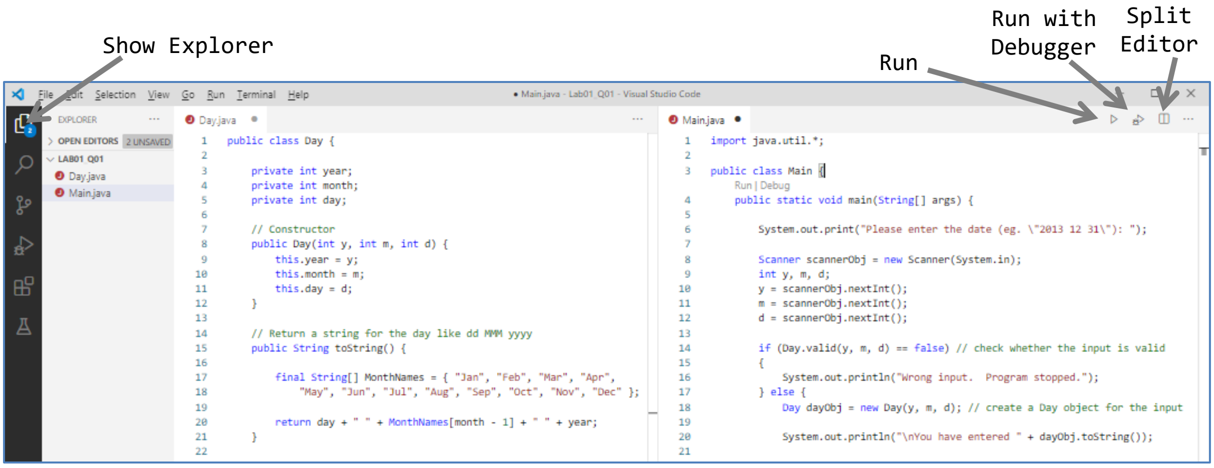


Figure 1

4.4 From the code in step 4.1, copy the contents of the `Main` and the `Day` classes into the source files. Use `the Split Editor button` to arrange the files in the window (Figure 2).



Day.java

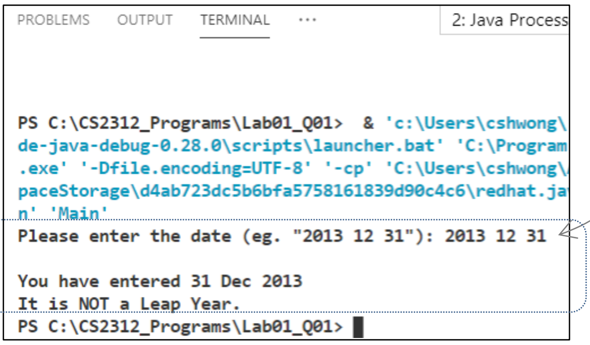
```java
public class Day {

    private int year;
    private int month;
    private int day;

    // Constructor
    public Day(int y, int m, int d) {
        this.year = y;
        this.month = m;
        this.day = d;
    }

    // Return a string for the day like dd MMM yyyy
    public String toString() {

        final String[] MonthNames = { "Jan", "Feb", "Mar", "Apr",
            "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };

        return day + " " + MonthNames[month - 1] + " " + year;
    }

```

Main.java

```java
import java.util.*;

public class Main {
    public static void main(String[] args) {

        System.out.print("Please enter the date (eg. \"2013 12 31\"): ");

        Scanner scannerObj = new Scanner(System.in);
        int y, m, d;
        y = scannerObj.nextInt();
        m = scannerObj.nextInt();
        d = scannerObj.nextInt();

        if (Day.valid(y, m, d) == false) // check whether the input is valid
        {
            System.out.println("Wrong input.  Program stopped.");
        } else {
            Day dayObj = new Day(y, m, d); // create a Day object for the input

            System.out.println("\nYou have entered " + dayObj.toString());

```

Figure 2

4.5 Execute the program by clicking `the Run button` above `Main.java` (Figure 2). Then, *enter the input* and observe the output.

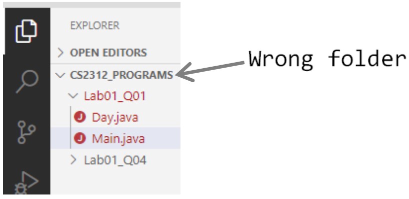


```
PS C:\CS2312_Programs\Lab01_Q01>  & 'c:\Users\cshwong\
de-java-debug-0.28.0\scripts\launcher.bat'  'C:\Program
.exe'  '-Dfile.encoding=UTF-8'  '-cp'  'C:\Users\cshwong\
paceStorage\d4ab723dc5b6bfa5758161839d90c4c6\redhat.ja
n'  'Main'
Please enter the date (eg. "2013 12 31"): 2013 12 31

You have entered 31 Dec 2013
It is NOT a Leap Year.
PS C:\CS2312_Programs\Lab01_Q01> █
```

**Troubleshooting:**

If you see an error message concerning ***package***, it is likely that you have opened a wrong folder.

Please check if it looks like this:



Explanation:

You should not open the **CS2312_Programs** folder because it is the root folder to contain all programs you will work on in this course.

Instead you should open the correct folder of **the program that you want to work with**. E.g. Lab01_Q01

To overcome:
Close the folder (File => Close Folder). Then check and follow steps 4.2 – 4.5 again.

4.6 Roughly read the code in the source files. See how much you can figure out about the program flow. The complete details will be explained in class.

**Q1.** Below is a copy of the sample program that you have tried in Step 4 in page 1-2.

**Your task**: Find out the brief meaning of each line in `Day.java` and `Main.java`

(Read *Q1_Explained.pdf* on the course web).

**Main.java**

```
import java.util.*;
public class Main
{
    public static void main(String[] args)
    {
       System.out.print("Please enter the date (eg. \"2013 12 31\"): ");
       Scanner scannerObj = new Scanner(System.in);
       int y, m, d;
       y=scannerObj.nextInt();
       m=scannerObj.nextInt();
       d=scannerObj.nextInt();

       if (Day.valid(y, m, d)==false) //check whether the input is valid
       {
         System.out.println("Wrong input.  Program stopped.");
       }
       else
       {
         Day dayObj = new Day(y,m,d); //create a Day object for the input
         System.out.println("\nYou have entered " + dayObj.toString());
         if (Day.isLeapYear(y))
             System.out.println("It is a Leap Year.");
         else
             System.out.println("It is NOT a Leap Year.");
       }
       scannerObj.close();
    }
}
```

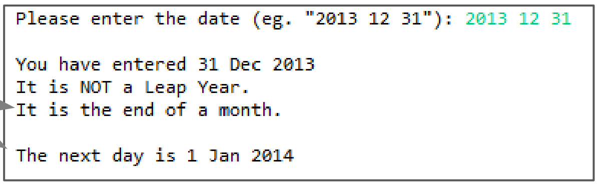
**Day.java**

```
public class Day {
    private int year;
    private int month;
    private int day;

    //Constructor
    public Day(int y, int m, int d) {
         this.year=y;
         this.month=m;
         this.day=d;
    }

    // Return a string for the day like dd MMM yyyy
    public String toString() {
        final String[] MonthNames = {
                "Jan", "Feb", "Mar", "Apr",
                "May", "Jun", "Jul", "Aug",
                "Sep", "Oct", "Nov", "Dec"};
        return day+" "+ MonthNames[month-1] + " "+ year;
    }

    // check if a given year is a leap year
    static public boolean isLeapYear(int y)
    {
      if (y%400==0)
          return true;
      else if (y%100==0)
          return false;
      else if (y%4==0)
          return true;
      else
          return false;
    }

    // check if y,m,d valid
    static public boolean valid(int y, int m, int d)
    {
      if (m<1 || m>12 || d<1) return false;
      switch(m){
          case 1: case 3: case 5: case 7:
          case 8: case 10: case 12:
                   return d<=31;
          case 4: case 6: case 9: case 11:
                   return d<=30;
          case 2:
                   if (isLeapYear(y))
                       return d<=29;
                   else
                       return d<=28;
      }
      return false;
    }
}
```

1.12 Finish online exercises for Q1 on Canvas.

**Q2**. You are to extend the program so that it

- checks if the date is the end of a month
- calculates the next date



```
Please enter the date (eg. "2013 12 31"): 2013 12 31

You have entered 31 Dec 2013
It is NOT a Leap Year.
It is the end of a month.

The next day is 1 Jan 2014
```

<u>Your tasks:</u>

2.1 Copy the **`Lab01_Q1`** folder as **`Lab01_Q2`** to do this exercise.
Close the Lab01_Q1 folder. Open the Lab01_Q2 folder.

2.2 Open the course web via **Canvas => CS2312 =>** https://www.cs.cityu.edu.hk/~helena/cs2312202021B

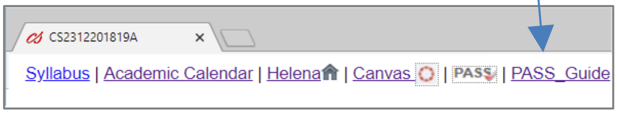
Add the given code from **`Q2_Given_Code.txt`**. Complete the program.

Learn the tricks in the hints <u>and finish the code quickly</u>.

2.3 Try several test cases on your own.

2.4 Submit to **PASS**. Make sure you get 100% correct.
Need help? Ask!

***Follow Page 1 of this user guide***


CS2312201819A
Syllabus | Academic Calendar | Helena | Canvas | PASS | PASS_Guide

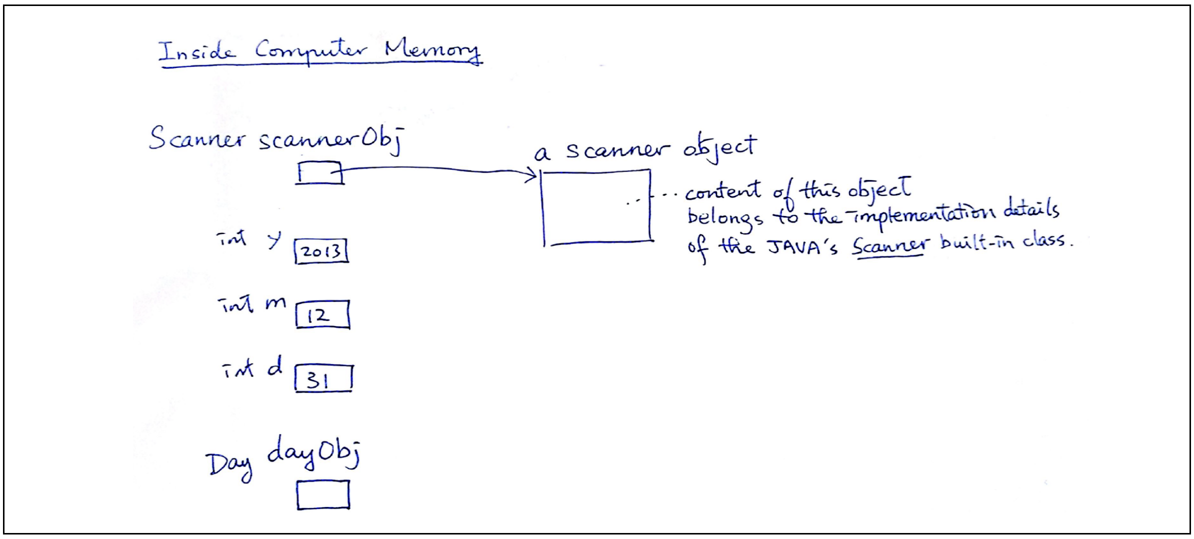
Canvas => CS2312 => https://www.cs.cityu.edu.hk/~helena/cs2312202021B

# Lab01- Progress Check

I. Understanding of terms / facts - Either put down a tick ☑ or ask the helper to teach you.)

- ☐ **class**
- ☐ **object**
- ☐ **data type**
- ☐ **variable**
- ☐ calling a method like: `xxx.xxx()`
- ☐ object variables are **references**, ie. they act like pointers
- ☐ static method vs non-static method; the different ways to call them

II. The following drawing is to show all variables of `main()`, as well as the objects created.
Your task: complete it for the `dayObj` variable and the `Day` object


Inside Computer Memory
Scanner scannerObj
a scanner object
content of this object belongs to the implementation details of the JAVA's Scanner built-in class.
int y
2013
int m
12
int d
31
Day dayObj

III. In C++, we have the *this* keyword that points to the object that a method is handling.
Similarly, in Java we have the *this* keyword that refers to the object.

Amy, Brian, and Carol write the Day constructors A, B, C on the right. All of them can compile and run correctly.

Your tasks: Read them and then circle the right answers below.

- Constructor A / B / C has poor parameter naming
- The uses of *this.* in A / B / C to refer to the year, month, day fields are **optional** and **can be removed** because there is no conflict of naming.

  For beginners: type the code and try!
- We can use *this.* in **static methods / non-static methods**.

  ↙ isLeapYear, valid ↙ toString

  For beginners: type the code and try!

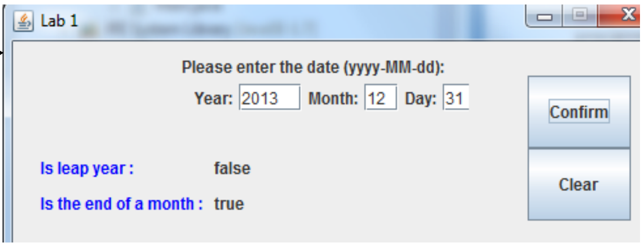
```
//Constructor (A) by Amy
public Day(int y, int m, int d) {
     this.year = y;
     this.month = m;
     this.day = d;
}
```

```
//Constructor (B) by Brian
public Day(int a, int b, int c) {
     this.year = a;
     this.month = b;
     this.day = c;
}
```

```
//Constructor (C) by Carol
public Day(int year, int month, int day) {
     this.year = year;
     this.month = month;
     this.day = day;
}
```

**Q3.** You are to experience with graphics mode programming, like this:

(using the JAVA tools: ***Swing*** and ***AWT***)



Please enter the date (yyyy-MM-dd):

Year: 2013 Month: 12 Day: 31

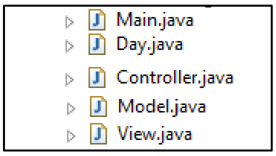
Is leap year : false

Is the end of a month : true

Your tasks:

3.1 Create a new program folder named `Lab01_Q3`

3.2 Download the following `.java` files from the course web and insert them to the folder:



Main.java
Day.java
Controller.java
Model.java
View.java

3.3 `Model.java` has something wrong.
It is because some methods are missing in `Day.java`.
Complete them according to the guidelines given at the end inside `Day.java`.

3.4 Test the program.

Roughly study the program source code.

The look and feel are controlled by the `MVC` modules (`controller.java, model.java, view.java`)

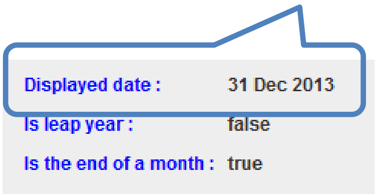
Your task: open each file and roughly take a look.

* This exercise is to give you a brief idea about such kind of graphics mode programs.
We will have no more graphics mode coding in CS2312. (**Not required for exam / quiz / midterm**)

If you want to know more about graphics mode programming with Swing/AWT and tools for designing/coding, you may google "tutorialspoint java swing awt"

3.5 Finish online exercises for Q3 on Canvas.

3.6 Suggested extension after you finish all Lab01: You may add the displayed date like this:
But note: **Finish Q4-6 first !!**



Displayed date : 31 Dec 2013

Is leap year : false

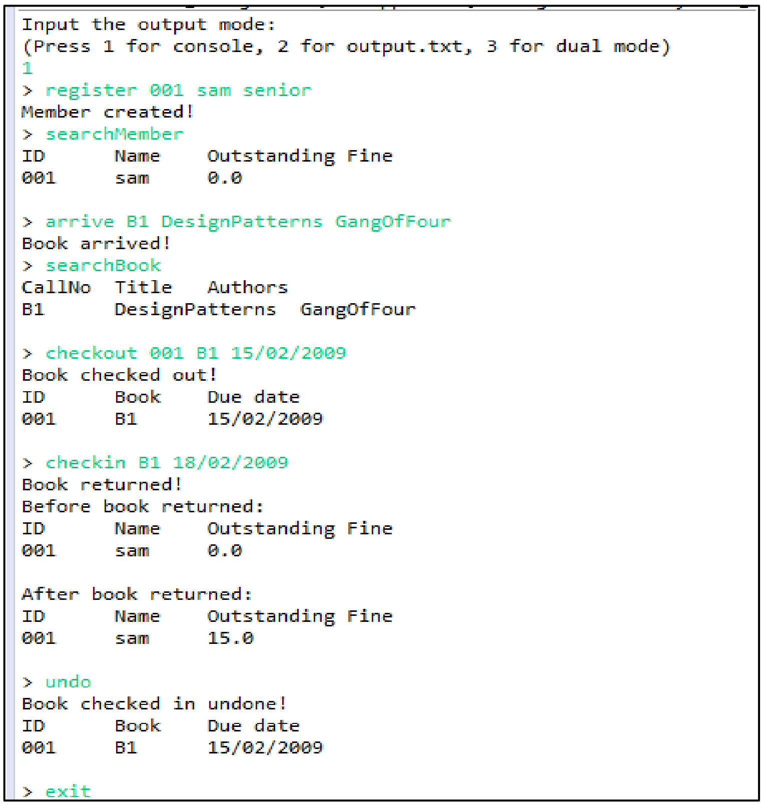
Is the end of a month : true

**Q4.** Examining an OO sample program: **`Library`**.

4.1 Open the link for this question from the course web => **Lab 01 *Q4 OO sample Library Program***
Follow the guidelines to run the program.

Test it as shown below:

Sample Rundown (Basic minimum requirement)



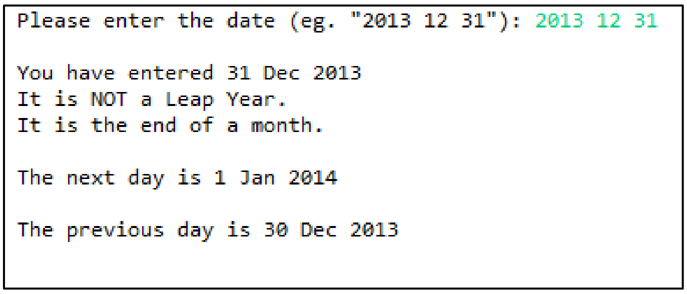
```
Input the output mode:
(Press 1 for console, 2 for output.txt, 3 for dual mode)
1
> register 001 sam senior
Member created!
> searchMember
ID      Name    Outstanding Fine
001     sam     0.0

> arrive B1 DesignPatterns GangOfFour
Book arrived!
> searchBook
CallNo  Title   Authors
B1      DesignPatterns  GangOfFour

> checkout 001 B1 15/02/2009
Book checked out!
ID      Book    Due date
001     B1      15/02/2009

> checkin B1 18/02/2009
Book returned!
Before book returned:
ID      Name    Outstanding Fine
001     sam     0.0

After book returned:
ID      Name    Outstanding Fine
001     sam     15.0

> undo
Book checked in undone!
ID      Book    Due date
001     B1      15/02/2009

> exit
```

4.2 Submit a testing rundown output to **Canvas** for marking (Simply copy-and-paste).
The above *Basic minimum requirement* is good enough.

You may do more if time is enough. *Well, this is to show that you have compiled and tried the library program. Therefore you do not need to do a very complete testing.*

Note also that CANVAS will not mark your work. Don't worry if you see a zero or a blank mark.

**Q5. [Take home exercise]** Extend your work for Q2 to show the previous day:



```
Please enter the date (eg. "2013 12 31"): 2013 12 31

You have entered 31 Dec 2013
It is NOT a Leap Year.
It is the end of a month.

The next day is 1 Jan 2014

The previous day is 30 Dec 2013
```

We will discuss the solution briefly during the coming lecture. But please try your best to complete it on your own first. ***Mind the deadline and Have fun!***

## Q6 OO Programming from C++ to Java

### C++ Version

You are given an OO programming question from a previous C++ exam paper.
The program models shops (class `Shop`) and customers (class `Customer`). Below is the `main` function. You may simply read the comments and quickly understand what it does:

Part (A) Program – the main function:

```
int main()
{
    Customer c1,c2;
    Shop s1,s2;

    c1.set(200); //c1 has $200 initially
    c2.set(200); //c2 has $200 initially

    s1.earn(c1, 30); //s1 earns and get $30 from c1
    s2.earn(c1, 40); //s2 earns and get $40 from c1
    s2.earn(c2, 50); //s2 earns and get $50 from c2

    cout << c1.getAmount() << endl; //expected output: 130
    cout << c2.getAmount() << endl; //expected output: 150
    cout << s1.getProfit() << endl; //expected output: 30
    cout << s2.getProfit() << endl; //expected output: 90

    return 0;
}
```

Part (B) Program
As an extension, students are to design the class `Group` that models a group of customers joined together for group-purchasing.

### JAVA Version

The above programs can be written using JAVA easily.
The Part (A) program only involves basic OO skills, that are not more than Lab01.
The Part (B) program requires using array of objects, that you will learn in Lab02.

Your Tasks

↙download the exam paper from course web, under Lab01 Given Files.

(1) Try to answer the questions of the C++ Version on paper.
(Yes you may skip this part because we only require JAVA in CS2312. But as a Computer Science student it helps you learn important foundation concept in C++, and then "Why JAVA", that is very likely to appear in an employment written test).

Discuss your answers with Helena or helpers. Or let us teach you. ^_^

(2) [**Compulsory**] Complete the JAVA version for both part (A) and (B):
Download Main.java from the course web and add all required classes.

Submit all .java files to PASS. Note that Main.java should not be changed.