Lab01 - Q1 Explanation

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

You have entered 31 Dec 2013
It is NOT a Leap Year.
```

① Include a library

② The Main class which contains the main method.

method = function

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)
        {
          System.out.println("Wrong input.  Program stopped.");
        }
        else
        {
          Day dayObj = new Day(y,m,d);
          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();
    }
}
```

Lab01 - Q1 Explanation

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

You have entered 31 Dec 2013
It is NOT a Leap Year.
```

⑤ static method is a method which does not operate on objects. ...learn later!

④ main() returns nothing (ie. return type is void)

③ simply a parameter for main(). It is needed but not used in most cases.

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)
        {
          System.out.println("Wrong input.  Program stopped.");
        }
        else
        {
          Day dayObj = new Day(y,m,d);
          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();
    }

}
```

Lab01 - Q1 Explanation

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

You have entered 31 Dec 2013
It is NOT a Leap Year.
```

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)
        {
          System.out.println("Wrong input.  Program stopped.");
        }
        else
        {
          Day dayObj = new Day(y,m,d);
          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();
    }

}
```

⑥ like C++'s cout

⑦ like C++

```
int y, m, d;
cin >> y;
cin >> m;
cin >> d;
```

⑧ Scanner is JAVA's built-in class for input streams.
To input from keyboard,
- we create a scanner object for the keyboard. (new, system.in)
- we set a variable to refer to the scanner object. (scannerObj, point to)
- then wait and read an integer from the scanner object. (nextInt())

For ⑧, ⑫ : In JAVA, an object variable is like a pointer in C++

Yes, we do not write * in JAVA

Lab01 - Q1 Explanation

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

You have entered 31 Dec 2013
It is NOT a Leap Year.
```

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)
        {
            System.out.println("Wrong input.  Program stopped.");
        }
        else
        {
            Day dayObj = new Day(y,m,d);
            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();
    }

}
```

⑨ The Day class in Day.java

⑩ a method in Day.java, which checks the input.

⑫ Use the Day class as a data type, set a variable (dayObj). Use the variable to refer to the Day object ... point to

⑪ Create a Day object

There are 3 fields y, m, d are passed to Constructor so that the constructor can set the 3 fields

Day.java

⑭ a method in Day.java

⑬ a method for Day objects, in Day.java

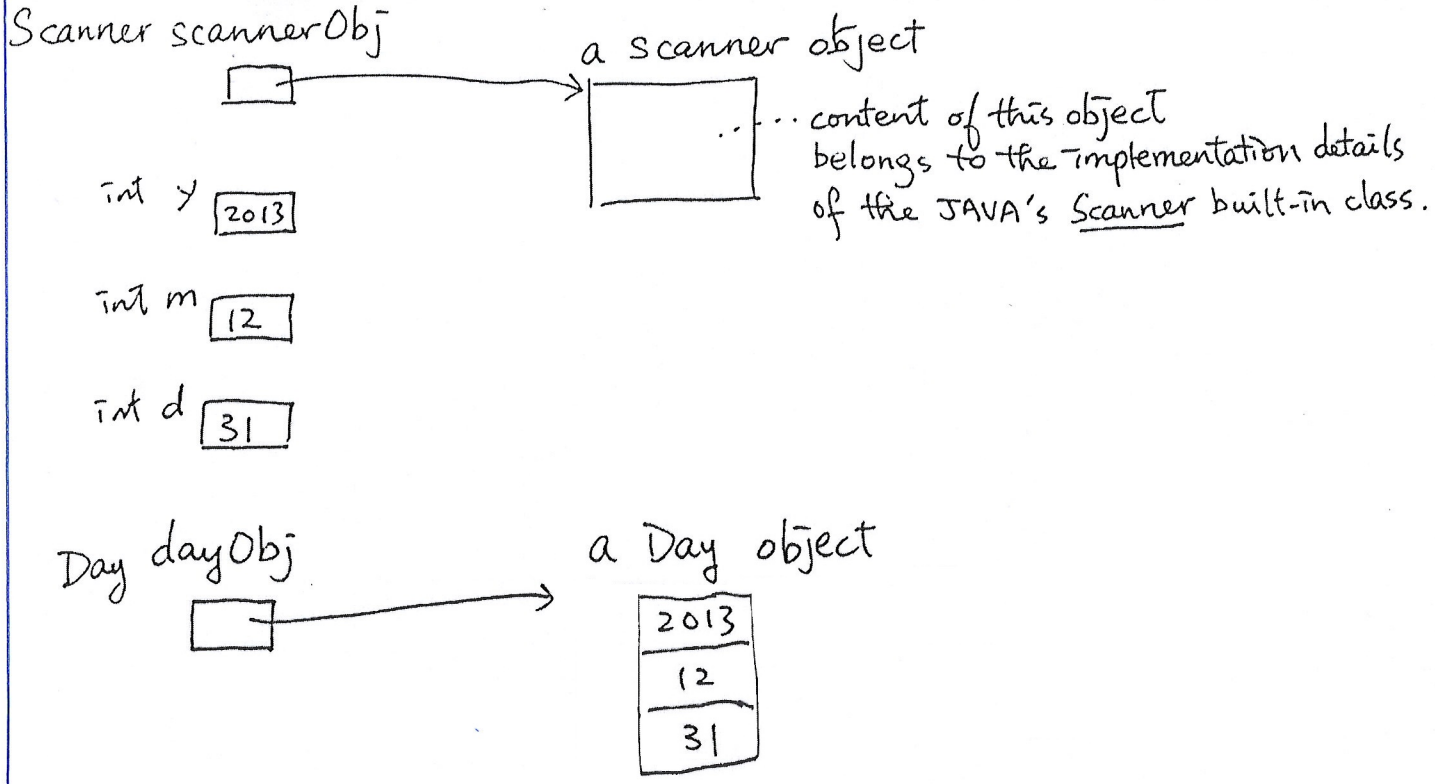
For ⑧, ⑫ : In JAVA, an object variable is like a pointer in C++

(Yes, we do not write * in JAVA)

Lab01 - Q1 Explanation

Variables of main(), and the created objects :

Inside Computer Memory



The Day Class

We write a class like class Day{...}
The class is a template for creating objects
We can use the class name as a data type for variables

class name

These variables act like pointers

Day.java

```
public class Day {

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

Define 3 fields for Day objects : year, month, day

```
    //Constructor ... called together with new
    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;
    }
}
```

Example
Main.java ... {dayObj.toString()} → year, month, day

"Const"

static methods are methods which do not operate on objects.

Examples
Main.java
{
Day.isLeapYear(y)
Day.valid(y,m,d)
}