# Lab 01 Introduction to Java and OO-basics Review

*[Preparation] Using VS Code in YEUNG-B7520*

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

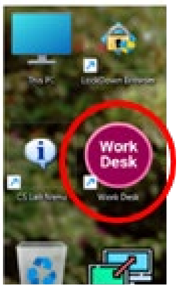
**[Tasks Q1-3 originally drafted by Mr. HO Wai Kit (BSC2 - 2014)]**

## [Preparation] Using VS Code in B7520 computers

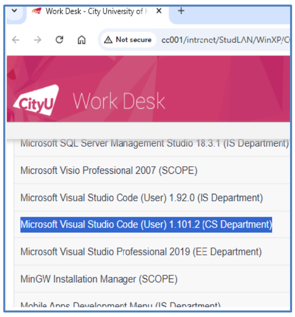
VS Code is an IDE (integrated development environment) that supports various coding languages. We will use it for writing Java Programs.

### Step 1 Start VS Code in a B7520 computer

1.1) Double-Click the Work Desk icon to open it in a browser.



1.2) Pick Microsoft Visual Studio Code (User) (CS Department)

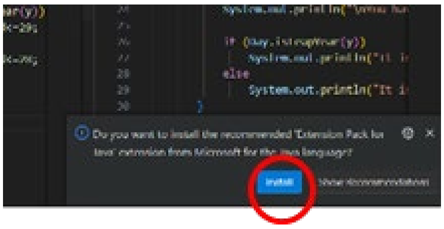

Work Desk - City University of H
Not secure
cc001/intranet/StudLAN/WinXP/C
CityU
Work Desk

Microsoft SQL Server Management Studio 18.3.1 (IS Department)
Microsoft Visio Professional 2007 (SCOPE)
Microsoft Visual Studio Code (User) 1.92.0 (IS Department)
Microsoft Visual Studio Code (User) 1.101.2 (CS Department)
Microsoft Visual Studio Professional 2019 (EE Department)
MinGW Installation Manager (SCOPE)
Mobile Apps Development Menu (IS Department)

1.3) Loading VS Code is automatic. It may take ~2 minutes …

Note:

At any time, when VS Code asks to install the ***Extension Pack for Java***. Please click ***Install***.



```
    if (Day.isLeapYear(y))
        System.out.println("It i
    else
        System.out.println("It i
}
```

Do you want to install the recommended 'Extension Pack for Java' extension from Microsoft for the Java language?

### Step 2 Prepare the program folder

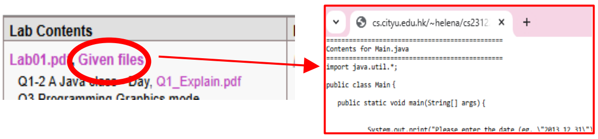
2.1) Create the folder for the CS2312 programs: `C:\CS2312_Programs`

2.2) Create the folder for the program of Lab 1 Q1: `C:\CS2312_Programs\Lab01_Q1`

### Step 3 Open the source code for Lab 1 Q1

3.1) Go to the course web: Canvas => CS2312 => https://www.cs.cityu.edu.hk/~helena/cs23122...

3.2) Open the source code for Q1:



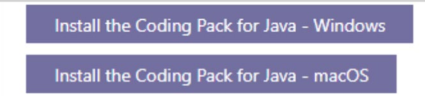
Lab Contents
Lab01.pdf, Given files
Q1-2 A Java class - Day, Q1_Explain.pdf
Q3 Programming Graphics mode

```
==================================================
Contents for 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\")
```

**To use your own computer** instead of the B7520 computers:

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

Please scroll down the page to find the following command buttons for installation:


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

Step 4 Create and run the sample Java Program:

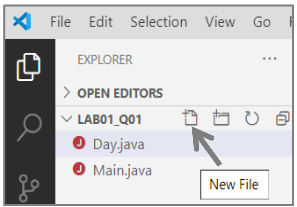


Figure 1

4.1 Recall: In Step 1 the program folder `C:\CS2312_Programs\Lab01_Q01` was created.

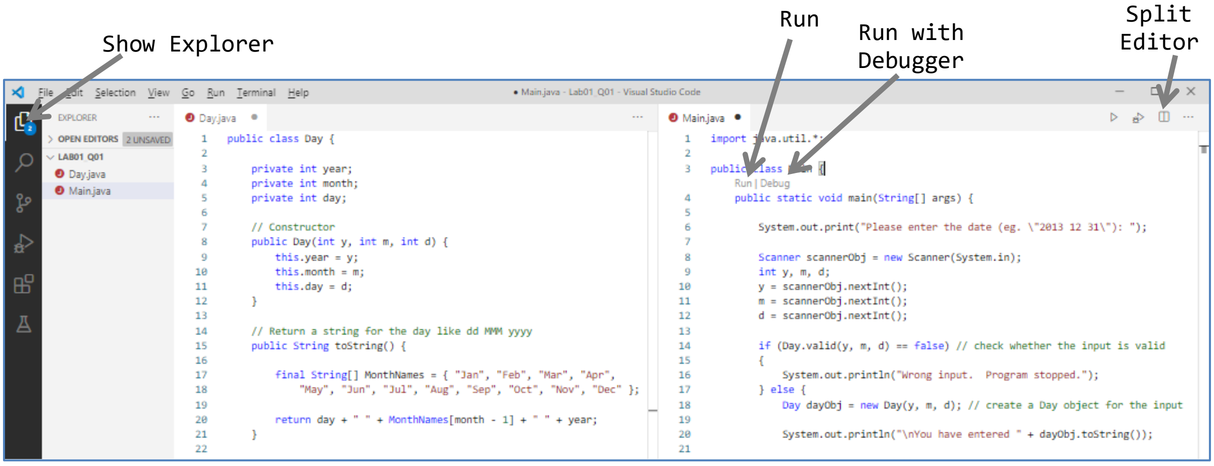
4.2 Now, open the program folder in VS Code:

`VS Code => File => Open Folder =>` select `C:\CS2312_Programs\Lab01_Q01`

4.3 Use the `New File` button to create two source files named `Main.java` and `Day.java` (Figure 1).

4.4 From the code opened in step 3, please copy the contents of the `Main` and the `Day` classes into the source files. Save the files.
Use `the Split Editor button` to arrange the files in the window (Figure 2).

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



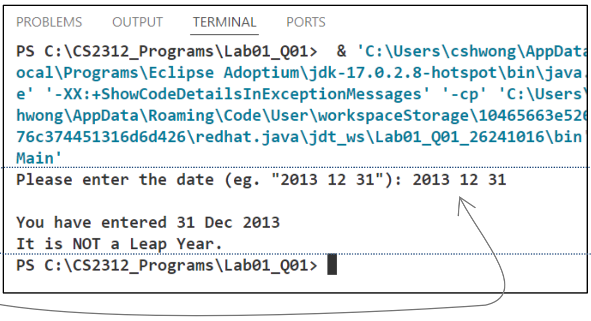
```java
public class Day {

    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;
    }

```

```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

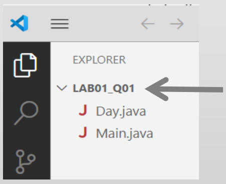
4.5 Click `the Run button` above `main` (Figure 2).
Then, *enter the input* and observe the output.

4.6 Roughly read Main.java and Day.java..
Try to figure out the program flow.

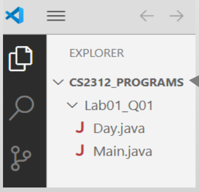


```
PS C:\CS2312_Programs\Lab01_Q01>  & 'C:\Users\cshwong\AppData
ocal\Programs\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.
e' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users
hwong\AppData\Roaming\Code\User\workspaceStorage\10465663e52
76c374451316d6d426\redhat.java\jdt_ws\Lab01_Q01_26241016\bin
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> █
```

**Caution 1:**
Always open exactly the program folder that contains the current program that you are now working on. It should contain just one single program only. Don't open the parent folder of the program folder.

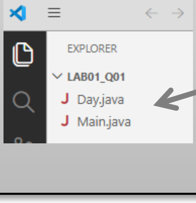


✓ Correct

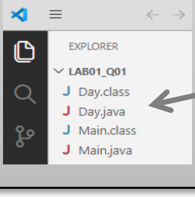


✘ Wrong folder!
That is, do not
open the parent
folder!!

**Caution 2:**
The program folder should have only the .java files. You should not see any .class files.



✓ Correct



✘ Incorrect!
You should not see
any .class files.

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

1.1. The file *Q1_Explained.pdf* (Canvas => CS2312 => https://www.cs.cityu.edu.hk/~helena/cs231220..), explains the detail meaning of each line in **Day.java** and **Main.java**. We will go through *Q1_Explained.pdf* in class.

**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**

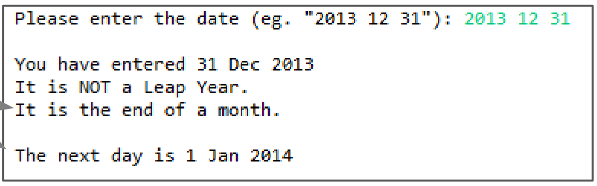
```
public class Day {
    private int year;
    private int month;
    private int day;

    //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;
    }
}
```

1.2 Online exercises for Q1 are given on Canvas. Please finish before the deadline.

1.3 For dangerous students (e.g., you are not confident on starting CS2312 smoothly), please redo the program on your own. That is, write Main.java and Day.java all from the beginning (You can find a demo video that teaches all details: Just view on https://www.cs.cityu.edu.hk/~helena/cs231220...)

**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
```

Your tasks:

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/cs23122021...**

Click the link named **Given Files** for Lab 1. You can see a folder of given files for Lab 1.

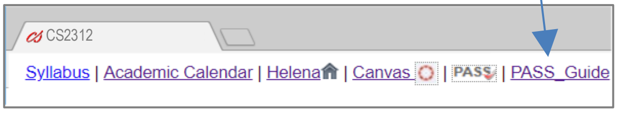
Add the given code from `Q2_Given_Code.txt`. Complete the program.

Learn the tricks in the hints and finish the code quickly.

2.3 Try several test cases on your own.

2.4 Submit to **PASS** https://pass3.cs.cityu.edu.hk/ .

Page 1 of this guideline shows you how to submit your programs onto PASS.


CS2312
Syllabus | Academic Calendar | Helena | Canvas | PASS | PASS_Guide

Canvas => CS2312 => https://www.cs.cityu.edu.hk/~helena/cs23122021...

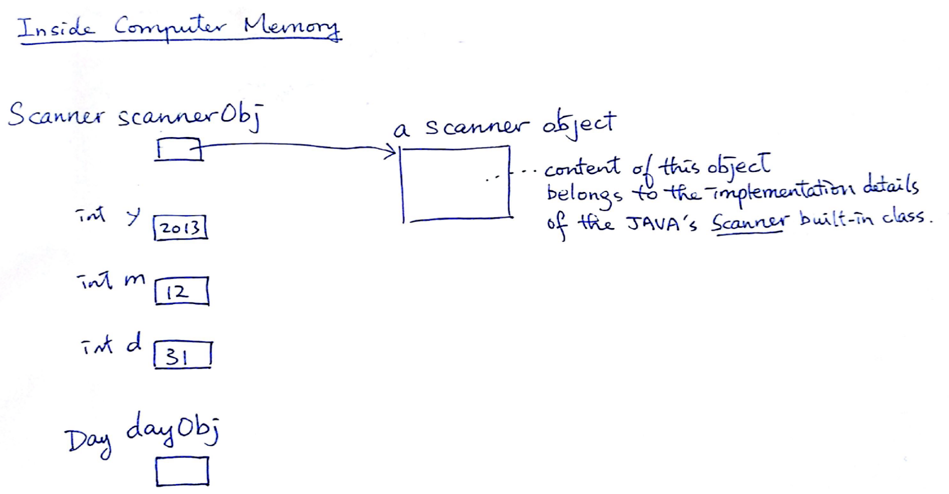
Make sure you get 100% correct.

# Lab01- Progress Check

I. Understanding of terms / facts - Either put down a tick ☑ or ask for help.)

- □ **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!
- ↙ isLeapYear, valid ↙ toString

  We can use *this.* in **static methods / non-static methods**.

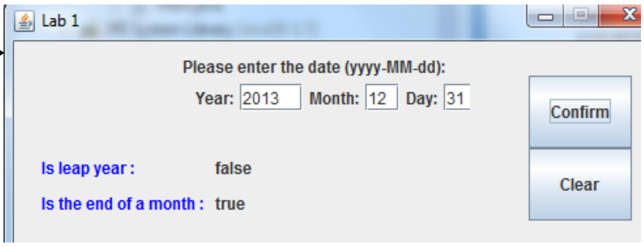
  For beginners: type the code and try!

```
//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:



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

Year: 2013 Month: 12 Day: 31

Is leap year : false

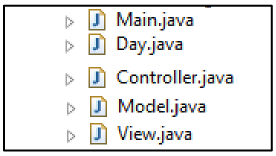
Is the end of a month : true

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

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 can 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"

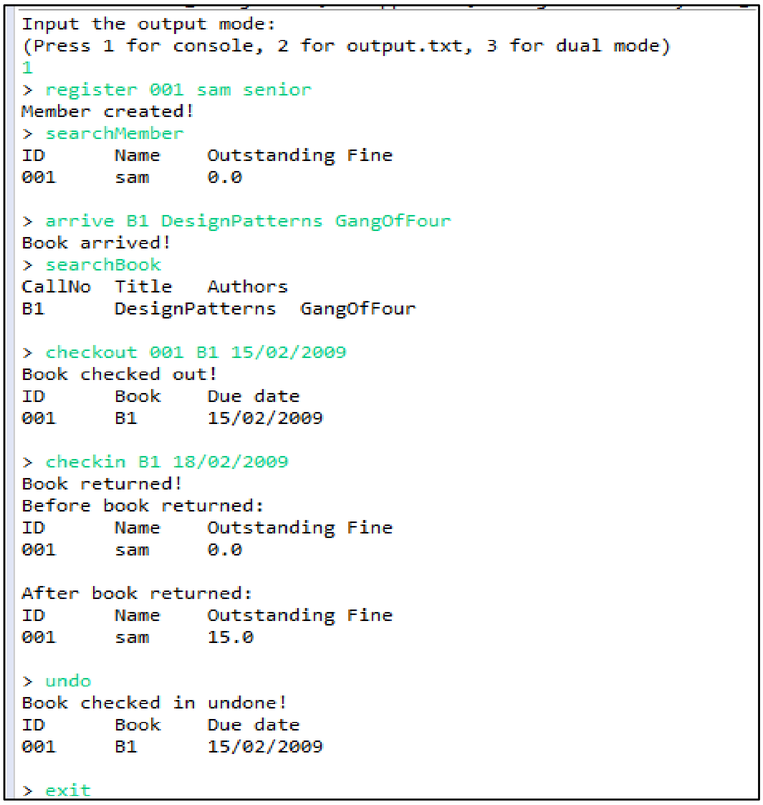
3.5 Finish the MCs for Q3 on Canvas.

**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)



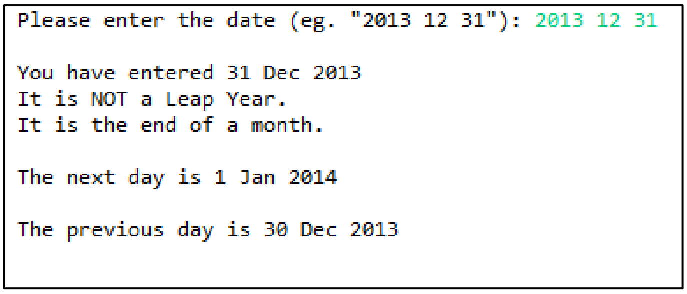
```
Input the output mode:
(Press 1 for console, 2 for output.txt, 3 for dual mode)
1
> register 001 sam senior
Member created!
> searchMember
ID      Name    Outstanding Fine
001     sam     0.0

> 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.** 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 lecture before the deadline. But please try to figure out the solution on your own first. ***Have fun!***

## Q6 OO Programming from C++ to Java

### Explanations:

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

You are given a C++ OO programming question from a previous C++ exam paper. (The paper is available on *https://www.cs.cityu.edu.hk/~helena/cs23122021... => Given Files for Lab 1*)

C++ Version

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;
}
```

However, the program has some problem and the question requires you to fix and improve it.

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: main() are given on next page

The above programs can be written using JAVA easily.
The Part (A) program only involves basic OO skills, that are not more than Lab01-Q1.
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).

(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.

Both (1) and (2) will be discussed briefly during the lectures before the deadlines. But students please first try to figure out the solutions on your own.

Q6 Part A (available at Canvas => CS2312 => Home => https://www.cs.cityu.edu.hk/~helena/cs231220...), under Lab01)

```
==========
public class Main{

    public static void main(String [] args)
    {
        Customer c1, c2;
        Shop s1,s2;

        c1 = new Customer(200);
        c2 = new Customer(200);
        s1 = new Shop();
        s2 = new Shop();

        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

        System.out.println(c1.getAmount()); //expected output: 130
        System.out.println(c2.getAmount()); //expected output: 150
        System.out.println(s1.getProfit()); //expected output: 30
        System.out.println(s2.getProfit()); //expected output: 90
    }
}
```

Q6 Part B (available at Canvas => CS2312 => Home => https://www.cs.cityu.edu.hk/~helena/cs231220...), under Lab01)

```
==========
public class Main{

    public static void main(String [] args)
    {
        Customer c1, c2;
        c1 = new Customer(200);
        c2 = new Customer(200);

        Group g1 = new Group();
        g1.add(c1);
        g1.add(c2);

        Shop s1 = new Shop();
        s1.earn(g1, 300); //s1 earns and get totally $300 from the group
        s1.earn(c1, 30);
        System.out.println(s1.getProfit()); //expected output: 330
        System.out.println(c1.getAmount()); //expected output: 20
    }
}
```