LABORATORY MANUAL

CE2002 / CZ2002 : Object-Oriented Design & Programming

SESSION 2016/2017

COMPUTER ENGINEERING COURSE
COMPUTER SCIENCE COURSE

SCHOOL OF COMPUTER SCIENCE & ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

OVERVIEW

1. LAB EXERCISES

The objective of laboratory experiments is to re-enforce your understanding on the course material and ***rely on your experiences and skills gained from prior level 1 courses***. As this is a level course, you will be expected to explore and discover means of completing the exercises – step-by-steps will be kept to the minimum. Some additional materials will be provided along for your references.
You may need to refer (cheatsheet/s) to other Java syntax like looping, branching, arrays etc to complete this exercise.

There are 4 lab. sessions altogether and the lab. sessions are conducted in the Computing Lab (N4-B2b-04) in SCE. The working programming environment runs on the Windows operating system.

Lab 1 – Windows Environment, My First Java Program, Fundamental Programming.
Lab 2 – Methods
Lab 3 – Classes & Objects, Array of Objects
Lab 4 – Inheritance and Polymorphism
Lab 5 – OO implementation in C++

The purpose of these exercises is for you to gain hands-on practical skill in developing Java program/application as well as using C++. Eventually, you will be required to complete an assignment using one of the OO languages.

2. EQUIPMENT

Hardware: Workstation running under the *Windows* environment.
Software: Eclipse IDE,
Java 2 Platform,
Microsoft Visual Studios 2010 (or above).

3. LAB ASSESSMENT *(IMPORTANT!)*

Upon completion of each lab session, you are to demonstrate the working of your work to the lab supervisor in charge. The lab supervisor may request that source codes are to be submitted. This will form part of the lab assessment. Any late submission/demonstration after the lab session will be penalized. Non-original work will also be penalized.

You are strongly encourage to check with your lab supervisor when in doubts and not wait till the last minute.

No make-up is allowed for students who have missed their stipulated lab classes without any acceptable excuse like having a valid leave of absence from the school or on medical leave. The students will be deemed to have failed the particular lab work. In case you have valid reasons to do makeup, you must inform the lecture coordinator in charge immediately. The makeup should be done within the same week, during the lab sessions attended by other groups.

4. ASSIGNMENT

You are also required to submit **an assignment** (to be annouced). You are required to work in groups (size to be determined later) and submit an report with your software deliverables. The assignment will also contribute to your final grade in this course.

The marking of your report is generally based on the following criteria:

1. Structure/organization/presentation of your Java codes;
2. Documentation of codes (whether the program is well commented to aid understanding);
3. Correctness of the program;
4. Thoroughness of the testing of the program;
5. Your understanding of the program;
6. User-friendliness of your program (how easy is it to use your program).

To achieve the above, you need to pay attention to:

1. Document your programs - This includes adding comments to your programs at appropriate locations, adding comments to state the purpose of the program, who the author is, and the time the program is written; in addition, you should also add comments on what is the purpose of each class and method.
2. Write reasonably indented codes and blank lines to enhance the readability of your programs.
3. Check all the test cases given to you to test the correctness of your programs.

Detail requirements for the assignment can be found on the website once the assignment is posted.

5. RULES OF CONDUCT

Please be reminded that **PLAGIARISM** (or copying part of/complete assignment) is considered as **CHEATING**, which is strictly prohibited. *Both the copier and code provider* may be **EXPELLED**.

Lab 1
WINDOWS ENVIRONMENT, MY FIRST JAVA PROGRAM & FUNDAMENTAL PROGRAMMING

1. OBJECTIVE

The objectives of Lab 1 are (1) to learn the basic IDE editor ; (2) to edit a program, (3) to compile and execute a program and (4) write working fundamental programs. This aims to help students to familiarize themselves with the working programming environment and fundamental Java syntax.

You can use either Eclipse IDE to develop your Java programs.
Appendix A & B give a guide on using the Eclipse IDE.

2. My First Java Program

In this exercise, one is going to write his/her first Java program by typing in a sample Java program *MyFirstProgram.java*. You may save your works in your preferred directory. Refer to Appendix B as a guide.

2.1 Create a Java Source File

Type in the following program *MyFirstProgram.java* in the editor window and then save the program in your home directory. Note that the program must be saved as *MyFirstProgram.java*.
[[In Java, the file name (at this instance, MyFirstProgram) must be the same as the class name]]

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

     System.out.println("Hello! This is my first program.");
     System.out.println("Bye Bye!")

  }
}
```

Figure 1

3. *Your Tasks for this LAB*

3.1 Write a program that reads a character from the user and then uses a *switch* statement to achieve what the following *if* statement does.

```
if  ((choice == 'A')  || (choice == 'a'))
       printf("Action movie fan\n");
else if  ((choice == 'C')  || (choice == 'c'))
       printf("Comedy movie fan\n");
else if  ((choice == 'D')  || (choice == 'd'))
       printf("Drama movie fan\n");
else
       printf("Invalid choice\n");
```

Important: Remember to name the source code of this program as **P1.java** and name the compiled class code as **P1.class** inside the sub-directory *lab1*.

Test cases: 'a', 'A', 'c', 'C', 'd', 'D', 'b', 'B'.

Expected outputs: 'a', 'A' – Action movie fan; 'c', 'C' – Comedy movie fan; 'd', 'D' – Drame movie fan; 'b', 'B' – Invalid choice.

3.2 The salary scheme for a company is given as follows:

Salary range for grade A: $700 - $899
Salary range for grade B: $600 - $799
Salary range for grade C: $500 - $649

A person whose salary is between $600 and $649 is in grade C if his merit points are below 10, otherwise he is in grade B. A person whose salary is between $700 and $799 is in grade B if his merit points are below 20, otherwise, he is in grade A. Write a program to read in a person's salary and his merit points, and displays his grade.

Important: Remember to name the source code of this program as **P2.java** and name the compiled class code as **P2.class** inside the sub-directory *lab1*.

Test cases: (1) salary : $500, merit : 10; (2) salaray : $610, merit : 5; (3) salary : $610, merit : 10; (4) salary : $710, merit : 15; (5) salary : $710, merit : 20; (6) salary : 800, merit : 30.

Expected outputs: (1) salary : $500, merit : 10 – Grade C; (2) salaray : $610, merit : 5 – Grade C; (3) salary : $610, merit : 10 – Grade B; (4) salary : $710, merit : 15 – Grade B; (5) salary : $710, merit : 20 – Grade A; (6) salary : 800, merit : 30 – Grade A.

3.3 Write a program to generate tables of currency conversions from Singapore dollars to US dollars. Use title and column headings. Assume the following conversion rate:

1 US dollar(US$) = 1.82 Singapore dollars (S$)

Allow the user to enter the starting value, ending value and the increment between lines in S$. The starting value, ending value and the increment are all integer values. Generate three output tables using the following loops with the same input data from the user:

1. Use a *for* loop to generate the first table;
2. Use a *while* loop to generate the second table; and
3. Use a *do/while* loop to generate the third table.

Place all the codes in the main() method.

Important: Remember to name the source code of this program as **P3.java** and name the compiled class code as **P3.class** inside the sub-directory *lab1*.

Test cases: (1) starting : 1, ending : 5, increment : 1; (2) starting : 0, ending : 40, increment: 5; (3) starting : 40, ending : 0, increment: 5 (treat this case as an error).

Expected outputs:
(1) starting : 1, ending : 5, increment : 1;

```
US$          S$
--------------
1            1.82
2            3.64
3            5.46
4            7.28
5            9.1
```

(2) starting : 0, ending : 40, increment: 5;

```
   US$       S$
   --------------
   0         0.0
   5         9.1
   10        18.2
```

```
15      27.3
20      36.4
25      45.5
30      54.6
35      63.7
40      72.8
```

(3) starting : 40, ending : 0, increment: 5 (treat this case as an error) – Error input!!

4.4 Write a program that reads the height from a user and prints a pattern with the specified height. For example, when the user enters height = 3, the following pattern is printed:

AA
BBAA
AABBAA

If the height is 7, then the following pattern is printed:

AA
BBAA
AABBAA
BBAABBAA
AABBAABBAA
BBAABBAABBAA
AABBAABBAABBAA

Important: Remember to name the source code of this program as **P4.java** and name the compiled class code as **P4.class** inside the sub-directory *lab1*.

Test cases: 0, 3, 7

Expected outputs: (1) height = 0 – Error input!! (2) height = 3 & (3) height = 7 – same as the sample patterns.