

## ASSIGNMENT 1

### 1. OBJECTIVE

The objective of this assignment is to practice your programming skills on branching, looping and methods. You need to submit this assignment for grading.

### 2. LABORATORY

This lab is conducted in the Computing Lab 1 (N4-b1-10) in SCE.

### 3. EQUIPMENT

Hardware: Workstation/PC running under the *Linux* environment.

Software: text editor

Java 2 Platform

### 4. EXPERIMENT

First create a sub-directory called *assign1* in your home directory. Do all the programming in this *assign1* sub-directory and leave all the programs in this *assign1*. Use the test cases suggested to test your program and understand why these test cases are used. You do not need to do (extra) error checking on input.

**4.1** Write a program to calculate the actual cost of buying a car in Singapore. Your program should input the list price and the category of a car, and print out the actual cost.

- Based on the list price, assuming that the car dealer will give a discount of 20%.
- You should also consider that there is a 10% luxury tax on the *excess* amount over \$100,000 (based on the discounted price).
- G.S.T. tax of 5% is also applied to the *discounted price*. Note that GST tax is not applied to the luxury tax.
- Certificate of Entitlement (COE) must be obtained for every car in Singapore. The amount you bid for the COE is based on the category your car belongs to. COE is not taxed. The categories and their COE prices are given in the following table:

Category (c.c.)	Price (\$)
(1) Car (1600 c.c. & below) & Taxi	20,000
(2) Car (above 1600 c.c.)	18,000
(3) Goods Vehicle & Bus	10,000
(4) Motorcycle	500

Your program should also handle error input. For example, if the list price is less than or equal to 0, then an appropriate error message will be output and no calculation will be done.

**Important:** Remember to name the source code of this program as **P1.java** and name the compiled class code as **P1.class** (this is automatically done after compilation) inside the sub-directory *assign1*.

**Test cases:** (1) price : \$35,000, category : 1; (2) price : \$150,000, category : 2; (3) price : \$30,000, category : 3; (4) price : \$5,000, category : 4; (5) price : \$0, category : 1 (give an error message);

(25 marks)

- 4.2** Write a Java program to read in a list of non-negative integers. Assume that the integers are entered one per line and a negative sentinel value (i.e. -1) is used to indicate the end of the input. The program will count the number of odd and even integers, and print the total counts and the average values for the odd and even input integers on the screen. You must **NOT** use **array** in your program.

**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 *assign1*.

**Test cases:** (1) 1 3 5 -1; (2) 2 4 6 8 10 -1; (3) 1 5 4 8 24 56 89 76 6 -1.

(25 marks)

- 4.3** Write a Java program that asks the user to input four non-negative numbers and draws the corresponding bar chart that is displayed vertically on the screen. For example, if the inputs are 5, 2, 4 and 3, then the following bar chart will be displayed. No error checking on input is required. You must **NOT** use **array** in your program.

```
**
**  **
**  ** **
** ** ** **
** ** ** **
** ** ** **
-----
```

**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 *assign1*.

**Test cases:** (1) 5, 4, 3, 2; (2) 2, 3, 4, 5; (3) 5, 3, 4, 2; (4) 2, 4, 3, 5; (5) 2, 4, 5, 3.

(25 marks)

- 4.4** Write a Java method **reverseSideDigits( )** which accepts a non-negative integer as its parameter value from a calling method, reverses the digits on either side of the middle of the integer, and returns the result back to the calling method. In addition, write a Java program to test the method **reverseSideDigits( )**. No error checking on the parameter of the method is required. You must **NOT** use **array** and **String** methods in this question.

Sample results of the method call for three different parameter values are tabulated below:

Value	Value returned from reverseSideDigits( )
13	13
1357	3175
13579	31597

The method header is given as follows:

```
public static long reverseSideDigits(long value)
```

**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 *assign1*.

**Test cases (for the parameter value in reverseSideDigits):**

(1) 1; (2) 45; (3) 4589; (4) 45689; (5) 456789.

**(25 marks)**

## 5. REPORT

- I. This assignment is due at 12.00 noon 26 September 2005 (Monday).
- II. For the report, one needs to submit hard copies of his/her in the following format:

For each part of the assignment, you need to write:

1. Pseudocode
2. Program logic
  - Describe the logic of your program and justify why you design your program in this way. For example, if you use while loop in your assignment, state whether you have used a counter-controlled while loop in your program and explain why you think counter-controlled while loop is suitable for your program.
3. Program listings with comments of P1.java, P2.java, P3.java and P4.java
4. The typescript file(s) to show the results of the testing of your programs using the test cases specified in this manual only

## 6. MARKING SCHEME

The marking of each part of assignment (i.e. 25 marks) will be based on the following criteria:

1. Pseudocode (5 marks)
2. Program logic (5 marks)
3. Java programs (15 marks)
  - presentation of the Java codes (whether the program is properly indented);
  - documentation of codes (whether the program is well commented to aid understanding); and
  - correctness of the program.

## 7. REFERENCE

[1] The textbook for CPE102/CSC102.