PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Department of Computer Engineering

Midterm Examination: Semester 2 **Academic Year**: 2013-2014

Date: 4th January, 2014 **Time**: 13:30 – **15:30 (2 hours)**

Subject Number: 241-211 **Room**: S201

Subject Title: Object Oriented Programming

Lecturer: Aj. Andrew Davison

Exam Duration: 2 hours

This paper has 3 pages.

Authorized Materials:

- Writing instruments (e.g. pens, pencils).
- Books (e.g. dictionaries) and calculators are **not** permitted.

Instructions to Students:

- *Answer questions in English.* Perfect English is **not** required.
- Attempt all questions.
- Write your answers in an answer book.
- Start your answer to each question on a new page
- Clearly number your answers.
- Any unreadable parts will be considered wrong.
- When writing programs, use good layout, and short comments; marks will not be deducted for minor syntax errors.
- The marks for each part of a question are given in brackets (...).

Question 1 (30 marks; 30 minutes)

a) Explain the differences between a *class* and an *object*? (13)

b) Explain *call-by-value* and *call-by-reference* parameter passing in Java. (12)

c) What is *modularization* and *abstraction*? (5)

Explain using words, diagrams, and small code fragments in your answers.

Question 2 (30 marks; 30 minutes)

a) Implement a class called Stats, which generates statistics information. (25)

After a Stats object is created, doubles can be added to it using the Stats.nextNumber() method, as shown below:

```
Stats s = new Stats();
s.nextNumber(1.1);
s.nextNumber(-2.4);
s.nextNumber(0.8);
```

Stats should include methods for the following:

- return the number of inputs (e.g. 3 for the example above);
- return the last number entered (e.g. 0.8 for the example above);
- return the sum of all the numbers entered (e.g. -0.5 for the example above);
- return the mean of all the numbers entered (e.g. -0.166667 (-0.5/3) for the example above);
- return the largest number entered (e.g. 1.1 for the example above)

Note that these methods can be called at any time, even if no numbers have been added to the Stats object with nextNumber(), and may be called again after more numbers have been added to the object.

Important: do **not** store all the number inputs in a data structure inside the Stats object. For example, do **not** use an array of doubles or an ArrayList to hold all the inputs inside the object.

b) Explain why it is a *good* idea to implement Stats without storing all the numbers inputs in the object. Also, explain why it is a *bad* idea. (5)

Question 3

(20 marks; 20 minutes)

a) Write a class called Point3D which can be used to store the position of a point in (x, y, z) 3D-space. (20)

For example:

```
Point3D pt1 = new Point3D(2.5, 1, 2.0);
```

creates a point which represents the coordinate (2.5, 1, 2.0).

Point3D should include methods for the following:

- change a point's coordinate (by supplying x, y, and z values, or by supplying an existing Point3D object);
- translate a point by a given (x, y, z) offset;
- return the current x-, y-, or z- value of a point;

Question 4

(40 marks; 40 minutes)

a) Write a Java program for a phone book, which stores an *ArrayList* of phone information. Each phone information object stores two pieces of data – the name of the person with that phone and their phone number. (20)
Hint: you should implement two classes: one for the phone book (e.g. a `PhoneBook` class), and one for phone information (e.g. a `PhoneInfo` class).

b) Write a main() method, clearly showing how you can add *and* remove phone information from a phone book. (10)

c) Could the phone book be implemented using a different data structure other than an ArrayList?

Explain your answer in words, with diagrams and small code fragments. (10)

--- End of Examination ---