UNIVERSITY TECHNOLOGY, MAURITIUS

MSc Software Engineering

Cohort: MSE/06/PT

Examinations for 2006 - 2007 / Semester 1

MODULE: DESIGN PATTERNS

MODULE CODE: SDT5101

Duration: 2 Hours

Instructions to Candidates:

1. Answer **all** questions.
2. Questions may be answered in any order but your answers must show the question number clearly.
3. Always start a new question on a fresh page.
4. All questions **do not** carry equal marks.
5. Total marks 60.

This question paper contains 3 questions and 3 pages.

ANSWER ALL QUESTIONS

QUESTION 1: (20 MARKS)

Referring to your class assignment which was about a University Management Information System. You were responsible to develop three sub-systems namely Staff Profiling System, Inventory Management System and Time-tabling systems. *(15 marks)*

(a) Discuss about any three design patterns you have used. Your answer should consist of a brief description of the pattern (including the structure) and a justification for its use and also how these patterns have been re-used. Note: There is no need to give codes to explain your answer but you can include diagrams in your discussion.

(b) What difficulties have you faced when designing and implementing the above mentioned systems using design patterns? A brief discussion will be sufficient. *(2 marks*

(c) Suppose that you have been assigned to re-engineer the existing systems for a company. Do you think that design patterns can be applied for these existing systems? Justify your answer. *(3 marks*

QUESTION 2: (20 MARKS)

(a) Describe the three categories of design patterns. Give two examples from each category. *(6 + 3 marks*

(b) When describing a pattern what is the purpose of "The Intent", and "The Collaborators" *(2 mark*

(c) What is an anti-pattern? Give two examples of anti-patterns. *(3 mark*

(d) What are the three reasons for using design patterns? *(6 mark*

QUESTION 3: (20 MARKS)

(a) What makes a Pattern a Pattern? *(5 m*

(b) Which pattern is shown below? Justify your answer. *(3 m*

```
public class A
{
        private static A instance = null;
        private A()
        {
        }
        public static A Createinstance()
        {
                if (instance == null)
                instance = new A();
                return instance;
        }
}
```

(c) Imagine that you need to write some program that needs to represent a building as rooms that can be manipulated. Manipulated as in interacting with objects in the room to change their state. The client that has ordered this program has determined that there will only be a need for a finite number of objects possible in each room, and a finite number of operations that can be performed on each of them. A sample action for a room is to "prepare it for a presentation". You have decided to use a design pattern which will deal with a large number of classes, but does not really need to bother the programmer with interacting with each of them when a room needs to be prepared. What pattern is most suitable for the situation given above? Justify your answer. Draw a diagram to illustrate the design pattern. *(6 n*

(d) People want to know patterns. What should their attitude be about patterns? How can people use patterns to do a better job? What is the real value? *(6 r*

*******END OF QUESTION PAPER******