

Seat No. _____



King Mongkut's University of Technology Thonburi

Final Exam of Second Semester, Academic Year 2015

COURSE CPE 231 Principles of Programming Languages

Computer Engineering Department, 2nd Yr.

Wednesday, May 11th, 2016

13.00-16.00

Instructions

1. This examination contains 4 problems, 8 pages (including this cover page).
2. The answers must be written in this exam paper. Please read the instructions carefully.
3. Calculator is not allowed.
4. Only one A4-sized note may be taken into the examination room.

Students will be punished if they violate any examination rules. The highest punishment is dismissal.

This examination is designed by

Marong Phadoongsidhi, Ph.D.

Tel. 0-2470-9083

The Final Exam

CPE 231 Principle of Programming Languages pg. 1/7

Student name:

Student ID:

Section 31 (International)

Instruction: This exam consists of 4 questions for a total of 100 points. You have 3 hours to work on them. You may bring in two A4 note sheets to the exam. Calculators not permitted.

Question 1 {30 pts. -- Concurrency}

(a) (10 pts) Describe relationships between computer programs, processes, and threads.

(b) (5 pts) Why has multithreading become an essential technique in modern software development?

The Final Exam

CPE 231 Principle of Programming Languages pg. 2/7

Student name:

Student ID:

Section 31 (International)

(c) (15 pts) Explain the Java thread states during its lifecycle, from the creation to termination of a thread.

The Final Exam

CPE 231 Principle of Programming Languages

pg. 3/7

Student name:

Student ID:

Section 31 (International)

Question 2 {20 pts. – Java Threads}

Consider the following threaded Java program :

```
public class SimpleThreads {

    public static void main(String args[]) throws InterruptedException
    {
        long patience = 1000 * 60 * 60; // 1 hr. (in milliseconds)
        threadMessage("Starting MessageLoop thread");
        long startTime = System.currentTimeMillis();
        Thread t = new Thread(new MessageLoop());
        t.start();

        threadMessage("Waiting for MessageLoop thread to finish");
        while (t.isAlive()) {
            threadMessage("Still waiting...");
            t.join(1000);
            if (((System.currentTimeMillis() - startTime) > patience)
                && t.isAlive()) {
                threadMessage("Tired of waiting!");
                t.interrupt();
                t.join();
            }
        }
        threadMessage("Finally!");
    }

    static void threadMessage(String message) {
        String threadName = Thread.currentThread().getName();
        System.out.format("%s: %s\n", threadName, message);
    }

    private static class MessageLoop implements Runnable {
        public void run() {
            String info[] = {"I", "Love", "Java", "Programming"};
            try {
                for (int i = 0; i < info.length; i++) {
                    Thread.sleep(2000);
                    threadMessage(info[i]);
                }
            } catch (InterruptedException e) {
                threadMessage("I wasn't done!");
            }
        }
    }
}
```

The Final Exam

CPE 231 Principle of Programming Languages

pg. 4/7

Student name:

Student ID:

Section 31 (International)

(a) (10 pts) What does the program print when executed?

(b) (10 pts) What is the printout if the variable *patience* is assigned a value 7000 ?

The Final Exam

CPE 231 Principle of Programming Languages pg. 5/7

Student name:

Student ID:

Section 31 (International)

Question 3 (25 pts. -- Exception Handling)

(a) (5 pts) What is an exception and why it can be useful to handle it? Can exception handling mechanism take care of both the runtime and logic errors? Why?

(b) (5 pts) Explain the try, catch, and throw exception handling process in Java.

The Final Exam

CPE 231 Principle of Programming Languages

pg. 6/7

Student name:

Student ID:

Section 31 (International)

(c) (15 pts) Suppose that statement2 causes an exception in the following try-catch block

```
try {  
    statement1;  
    statement2; // causes exception  
    statement3;  
}  
catch (Exception1 ex1) {  
}  
catch (Exception2 ex2) {  
}  
statement4;
```

Answer the following questions:

i) Will statement3 be executed?

ii) If the exception is not caught, will statement4 be executed?

iii) If the exception is caught in the catch block, will statement4 be executed?

iv) If the exception is passed to the caller, will statement4 be executed?

v) How can you make sure statement4 is always executed

The Final Exam

CPE 231 Principle of Programming Languages

pg. 7/7

Student name:

Student ID:

Section 31 (International)

Question 4 {25 pts. -- Functional Programming}

(a) (5 pts) Briefly describe the difference between the imperative and declarative programming paradigms.

(b) (5 pts) List one benefit of functional programming. Explain.

(c) (5 pts) What are Curried functions and why they may be useful? Give simple example.

(d) (10 pts) Show the output of the following Scala code:

```
def identity(x: Int) = x
def sqr(x: Int) = x * x
def sum(f: Int=>Int, a: Int, b: Int): Int =
  if (a == b) f(a) else f(a) + sum(f, a + 1, b)
println(sum(identity, 1, 5))
println(sum(sqr, 1, 5))
```