Rochester Institute of Technology

Department of Computer Science

CSCI-605, Spring ’23

Final Exam

Professor Eduardo Lima

Please make sure you have 9 pages in your exam including this cover page, printed single-sided. The last page is blank.

Provide complete and concise answers. Try to limit yourself to the space below the question.

If you need more space, use the last page.

This exam has 5 questions, each worth 8 points, totalling 40 points.

Student Full Name: ____

Question 1

```java
package exam3;
import java.util.*;
import static java.lang.Thread.sleep;

class Drop<T> {
    private Queue<T> messages = new LinkedList<>();
    private Integer MAX = 10;

    public synchronized T take() {
        while (messages.size()==0) { try { wait(); } catch (InterruptedException e) {} }
        try{ sleep(1000); } catch (Exception e) {};
        notifyAll();
        T message = messages.remove();
        System.out.printf("DROP: %s%n", messages);
        return message;
    }

    public synchronized void put(T message) {
        while (messages.size()==MAX) { try { wait(); } catch (InterruptedException e) {} }
        try{ sleep(1000); } catch (Exception e) {};
        messages.add(message);
        System.out.printf("DROP: %s%n", messages);
        notifyAll();
    }
}

class Producer implements Runnable {
    private Drop<String> drop;
    public Producer(Drop drop) { this.drop = drop; }

    public void run() {
        Random random = new Random();
        while(true) { drop.put(String.valueOf(random.nextInt(100))); }
    }
}

class Consumer implements Runnable {
    private Drop<String> drop;
    public Consumer(Drop drop) { this.drop = drop; }

    public void run() {
        Random random = new Random();
        while (true) { drop.take(); }
    }
}

public class DropTest {
    public static void main(String[] args) {
        Drop<String> drop = new Drop<>();
        for (int i=0; i<4; i++) {
            (new Thread(new Producer(drop))).start();
            (new Thread(new Consumer(drop))).start();
        }
    }
}
```

Answer the following (questions are in next page).

Question 1

1. How many producer threads will be created?

2. How many consumer threads will be created?

3. How many messages can be stored in a Drop instance?

4. Which object(s) is(are) acting as lock(s)/monitor(s) in this program?

Question 2

Server.java:

```
package exam3;
import java.net.*;
import java.io.*;

class ServerThread extends Thread {
    private Socket socket;
    public ServerThread(Socket socket) { this.socket = socket; }

    public void run() {
        try ( PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
              BufferedReader in = new BufferedReader(
                      new InputStreamReader( socket.getInputStream()));) {
            String inputLine;
            out.println("begin");
            while ((inputLine = in.readLine()) != null) {
                if (inputLine.equals("bye")) break;
                out.println(2*Integer.parseInt(inputLine));
                System.out.println("got"+inputLine);
            }
            socket.close();
        } catch (IOException e) {}
    }
}

public class Server {
    public static void main(String[] args) throws IOException {
        try (ServerSocket serverSocket = new ServerSocket(12345)) {
            while (true) { new ServerThread(serverSocket.accept()).start(); }
        } catch (IOException e) {}
    }
}
```

Client.java:

```
package exam3;
import java.io.*;
import java.net.*;

public class Client {
    public static void main(String[] args) throws IOException {
        try ( Socket socket = new Socket("localhost", 12345);
              PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
              BufferedReader in = new BufferedReader(
                      new InputStreamReader( socket.getInputStream()));) {
              BufferedReader stdIn = new BufferedReader( new InputStreamReader( System.in));
            in.readLine().equals("begin");                                                    //1
            while(true) {
                out.println(stdIn.readLine());
                System.out.printf("Server Response: %s%n", in.readLine());
            }
        } catch (IOException e) {}
    }
}
```

Answer the following (questions are in next page).

Question 2

1. Describe what this program does.

2. If we remove the statement with the in-line comment “1”, what will happen?

3. If multiple clients connect to the server at the same time, will the server be able to accept more than 1 connection? Why?

4. A socket is created when a connection is established, and stores a port at the server side and a port at the client side. In this program, what port is being used in the server side? What port is being used in the client side?

Question 3

```
package exam3;

public class Deadlock extends Thread {
        private int id;
        static Object a = new Object(), b = new Object();

        public Deadlock(int id) {
            this.id = id;
            if ( id == 0 ) ( new Deadlock(1) ).start();
        }

        public void run () {
            for (int i=0; i<2; i++) {
                if (id == 0) {
                    synchronized (a) {
                        try { sleep(100); } catch (Exception e) {}
                        synchronized (b) {
                            System.out.println(id);
                            try { sleep(100); } catch (Exception e) {}
                        }
                    }
                } else {
                    synchronized (a) {
                        try { sleep(100); } catch (Exception e) {}
                        synchronized (b) {
                            System.out.println(id);
                            try { sleep(100); } catch (Exception e) {}
                        }
                    }
                }
            }
        }
        public static void main (String[] args) { new Deadlock(0).start(); }
    }
```

Answer the following questions.

1. What are the possible outputs of this program?

2. Will this program ever run into a deadlock? Why?

Question 4

```java
package exam3;

public class Threads extends Thread {
    static Object lock;
    int id;

    public Threads(Object lock, int id) { this.lock = lock; this.id = id; }

    public void run () {
        System.out.println(id + " in");
        synchronized (lock) {
            try{ sleep(100); } catch (Exception e) {}
        }
        System.out.println(id + " out");
    }

    public static void main(String[] args) {
        Thread a = new Threads(new Object(), 1);
        a.start();
        Thread b = new Threads(new Object(), 2);
        b.run();
    }
}
```

Answer the following question.

1. Which possible outputs this program has?

Question 5

```
package exam3;
import java.util.function.Function;
import java.util.function.Predicate;
import java.util.function.Supplier;
import java.util.ArrayList;
import java.util.List;

public class Lambda {
    public static void main(String[] args) {
        Function<Integer, String> function = Integer::toBinaryString;                    //1
        System.out.printf("Binary representation of %s is %s%n", 7, function.apply(7));

        Supplier<List<Integer>> supplier = () -> new ArrayList<>();                      //2
        List<Integer> list = supplier.get();

        Predicate<List> predicate = new Predicate<>() {                                  //3
            public boolean test(List list) { return list.isEmpty(); } };                 //4
    }
}
```

Answer the following.

1. Write the equivalent lambda expression to the statement with in-line comment “1”

2. Write the equivalent method reference to the statement with in-line comment “2”

3. Write the equivalent lambda expression to the statement with in-line comments “3” and “4”

blank page