

CSCI 201L Final – Written SOLUTION
Fall 2014
12% of course grade

- 1. Interfaces and Inheritance** – Does the following code compile? If so, what is the output? If not, why not? Explain your answer. **(1.5%)**

```
interface I2 {
    public void meth1();
}
interface I1 extends I2 {
    public void meth1();
}
class C1 implements I1 {
    public void meth1() {
        System.out.println("2");
    }
}
class C2 extends C1 implements I2 {
    public void meth1() {
        System.out.println("1");
    }
}
public class Question1 {
    public static void main(String[] args) {
        I1 i1 = new C1();
        i1.meth1();
        I2 i2 = new C2();
        i2.meth1();
        C1 c1 = new C2();
        c1.meth1();
    }
}
```

0.5% - The code does compile.

0.5% - The output is

2

1

1

0.5% The runtime type of the object will determine which method gets called.

- 2. Network Addresses** – Assume that your employer decided to disable DHCP and instead have each employee assigned a static IP address. Your laptop is assigned IP address 192.168.2.5 with no subnets.

At home, you have DHCP enabled on your network. You forgot to change your static IP address from work one day on your laptop when you got home, and you didn't have network access. You checked your other computer (which does have network access), and you saw the IP address on your other computer on your home network was 192.168.1.3 with no subnets. Explain why your laptop does not have network access on your home network. **(1.5%)**

192.168.2.5 = 1100 0000 1010 1000 0000 0010 0000 0101

192.168.1.3 = 1100 0000 1010 1000 0000 0001 0000 0011

0.5% - These are Class C addresses.

1.0% - These IP addresses are in different networks. A Class C address uses 24 bits for the network address, which would be the first three octets. Since the first three octets of each IP address do not match (192.168.2 and 192.168.1), that means that the IP addresses are in different networks. An IP address in one network will not be able to communicate when it is in a different network.

3. Serialization – Does the following code compile? If so, what is the output? If not, explain why. (0.5% + 1.0%)

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;

class SerializeMe implements Serializable {
    public static final long serialVersionUID = 2;
    public transient String s1 = "csci201";
    public String s2 = "csci103";
    public int num = 3;
    public void print() {
        System.out.println(serialVersionUID);
        System.out.println(s1);
        System.out.println(s2);
        System.out.println(num);
    }
}

public class Question3 {
    public static void main(String[] args) {
        try {
            SerializeMe sm1 = new SerializeMe();
            sm1.print();
            FileOutputStream fos = new FileOutputStream("output.txt");
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(sm1);
            oos.close();
            FileInputStream fis = new FileInputStream("output.txt");
            ObjectInputStream ois = new ObjectInputStream(fis);
            SerializeMe sm2 = (SerializeMe)ois.readObject();
            ois.close();
            sm2.print();
        } catch (IOException ioe) {
            System.out.println("IOE: " + ioe.getMessage());
        } catch (ClassNotFoundException cnfe) {
            System.out.println("CNFE: " + cnfe.getMessage());
        }
    }
}
```

0.5% - The code does compile.

1.0% - The output is

2

csci201

csci103

3

2

null (NOTE: If they have csci201 here, deduct 0.5%)

csci103

3

4. Monitors and Locks – When Java uses the **synchronized** keyword, it is utilizing monitor functionality. There is also Lock functionality in Java through the Lock interface. Is the functionality behind a monitor the same as that through an explicit Lock variable? Why would a programmer choose to use a Lock instead of a monitor? **(1.0% + 1.0%)**.

1.0% - A monitor consists of a lock, but we do not have access to Conditions on the underlying lock.

1.0% - A programmer would choose to use a Lock when Conditions are needed.

0.5% if they say that a programmer could also choose to use a Lock for personal preference.

- 5. Semaphores** – We learned three different ways to implement critical sections of code with concurrent programming – monitors, locks, and semaphores. Assume one of your friends tells you about this new programming language called JavaSem. It has all of the functionality of Java but no monitors and locks, just semaphores. Would you consider this a limitation to the language or would JavaSem be able to implement all of the functionality of Java? Explain your answer. **(2.0%)**

1.0% - Semaphores can be used to implement monitor and basic lock functionality by setting the number of permits to 1 (which is also called a binary semaphore).

1.0% - We do not have Conditions on semaphores though, so we would need to utilize additional data to implement this functionality. Students can either propose something here (such as using a semaphore with no permits to force waiting), but that would require much thought. Grade this part leniently.

Other answers may be acceptable here as well, but the above are the two most important reasons.

6. **Distributed Programming** – RMI, CORBA, and Web Services all have advantages and disadvantages, which is why all three of them still exist. Assume you are working at a company who does not have any distributed application currently but wants to implement one because of the heavy processing power needed to factor large numbers. Give three design decisions that you would need to take into consideration to determine which distributed computing paradigm to use, along with the paradigm to use based on the decision. (0.5% + 0.5% + 0.5%)

Here is a sample answer: The language of the client and server should be taken into consideration. If both can be written in Java, RMI should be used. If they may be in different languages, CORBA or Web Services should be used.

0.5% - If the network administrators do not want to open an additional port through the firewall, Web Services should be used since it operates over the standard HTTP port, whereas RMI and CORBA do not.

0.5% - If the data being transmitted is very complicated with many objects, RMI may want to be used to reduce the details of mapping objects in different languages down to IDL.

0.5% - If the server will be accessed from outside the organization, CORBA or Web Services may be used because they allow you to implement explicit security policies.

Other answers will be acceptable as well.

7. **Threads** – Give three possible outputs of the following code? What are two rules that will always be true for the output (i.e. what will always be printed before or after something else)? (1.0% + 1.0%)

```
class T extends Thread {
    private int num;
    public T(int num) {
        this.num = num;
    }
    public void run() {
        System.out.println(num + "a");
        Thread.yield();
        System.out.println(num + "b");
    }
}
public class Question7 {
    public static void main(String[] args) {
        T t[] = new T[4];
        for(int i=0; i < 4; i++) {
            t[i] = new T(i);
            t[i].setPriority(Thread.MIN_PRIORITY + i);
        }
        for(int i=0; i < 4; i++) {
            t[i].start();
        }
    }
}
```

Outputs – these are three sample outputs, but as long as the outputs correspond to the rules below, they will be accepted (0.4% for the first output, 0.3% for each of the other two outputs)

```
3a    3a    2a
3b    2a    3a
2a    1a    2b
2b    0a    3b
1a    0b    1a
1b    1b    0a
0a    2b    1b
0b    3b    0b
```

Rules – the following two rules are the only rules.

0.5% - The “a” output for a specific number will always be printed before the “b” of a specific number.

0.5% - There will always be 8 outputs.

This is something that we will give credit for even though it is not a rule:

0.5% - The priority will not necessarily be acknowledged, but the higher numbered threads (which have higher priorities) will most likely be printed before the lower numbered ones.