

**Software Engineering  
General Examination  
March 17, 2004**

**Answer five of the seven questions.**

1. Answer each of the following:

- Compare and contrast the waterfall life cycle model and the prototyping model.
- Compare and contrast the object-oriented model and the spiral model.
- Order the following tasks in terms of the waterfall model: acceptance testing, project planning, unit testing, requirements review, cost estimating, high-level design, low level design, systems testing, design review, implementation, requirement specification.

2. Answer each of the following:

- Why is it hard to measure complexity?
- Calculate McCabe's cyclomatic number on the following source code using at least two methods. Draw the control flow graph.

```
cin >> a >> b >> c;
if (a > 10)
{
    cout << "hello";
    if (b < a)
    {
        cout << "part 1";
        if (c > a)
        {
            cout << "part 2";
        }
    }
    else
    {
        cout << "part 3";
    }
}
cout << "exiting";
```

3. Answer each of the following:

- Why are mathematical notations, such as sets, good foundations for specifications?
- What is the difference between preconditions, post-conditions, and invariants?
- Provide an argument that suggests the using formal specifications is not a good idea.
- Provide an argument that supports the use of formal specifications.

4. Answer each of the following:

- Is the following statement true or false? Explain.  

“A Petri net representation can always be represented as an equivalent finite state machine, but a finite state machine cannot always be represented as an equivalent Petri net.”
- Define a Pert diagram? What is the advantage of using a PERT diagram? How does a delay on the critical path affect the project? What is slack time? Why is slack time based on the earliest of the latest start times of the successor tasks?

5. Construct a finite state machine that recognizes the following set of strings:

$ad^n(cd)^me$  where  $n \geq 1$  and  $m \geq 1$ .

6. Answer each of the following:

- Define a data flow diagram. Include the definition of all of the components of a data flow diagram.
- What does leveling a data flow diagram mean?
- What to data flow diagrams specify about control?
- Construct a data flow diagram that represents the ordering of a meal at a restaurant.

7. Answer each of the following:

- Given the following code, draw the Control Flow Graph and generate a minimal set of test cases that meet each of the following criteria:

- Statement coverage
- Branch coverage

```
read hr1 min1 AmOrPm1
read hr2 min2 AmOrPm2
if (hr1 == 12)
    hr1 = 0
if (hr2 == 12)
    hr2 = 0
if (AmOrPm1 == pm)
    hr1 = hr1 + 12
if (AmOrPm2 == pm)
    hr2 = hr2 + 12
if (min2 < min1)
    min2 = min2 + 1
    hr2 = hr2 - 1
if (hr2 < hr1)
    hr2 = hr2 + 24
elapsed = min2 - min1 + 60 * (hr2 - hr1)
print elapsed
```

- What is path testing and is it practical? Explain.
- What is exhaustive testing and is it practical? Explain.
- If a program has two integer inputs and each can be a 32-bit integer, how many possible inputs does this program have?