Software Engineering, Mid-Term Practice Questions

1. What are the activities involved in the *waterfall* process? Also, what are the inputs and outputs of each activity?

2. What do we mean by *low representational gap*?

3. What type of UML diagram would you use to sketch a *domain model*?

4. In UML notation, how would you distinguish (visually) between `extends` and `implements`? (draw a picture)

5. What does a strong composition relationship look like in UML (sketch a diagram)? For example a `Car` has-an `Engine`.

6. Name one of each of the following: a Scrum role, a Scrum artifact, and a Scrum event.

7. In Scrum, who is responsible for prioritizing the requirements?

8. What is supposed to happen at the daily standup?

9. Name any one of the four *values* that are part of the manifesto for agile software development.

10. What is the central idea of Test-Driven Development?

11. Write a brief use case for a user who wants to make a photocopy with a copy machine. Identify and include a main success scenario, an extension, and a precondition.

12. Write a regular expression that matches names containing an optional middle initial. At minimum, your regex should match strings like `John Public`, `John Q. Public`, and `Mary L Smith`. Also, your regex should *not* match `JohnPublic`, `James MSmith`, or `john public`.

13. What is *grouping* in regular expressions and what is the syntax for grouping?

14. Name a specific example of each of the following kinds of patterns: *creational*, *behavioral*, and *structural*.

15. Which design pattern makes a family of algorithms interchangeable at run-time?

16. Draw a UML sequence diagram that shows possible interactions between objects in the *Observer* pattern. Your diagram should reference the client application, an observable, and an observer. Your diagram should also show: 1) how an observer subscribes to an observable and; 2) how an event causes an observer to update its dependents.

17. Fill in the following class `Singleton` to make it a correct Java implementation of the *singleton* pattern. Use the extra whitespace as hints for filling in keywords, access modifiers etc.

```
public class Singleton {
                  Singleton uniqueInstance;

           Singleton() {}

                 Singleton getInstance() {
      if (                       ) {
         uniqueInstance =
         }

      return
      }
}
```

18. Complete the following implementation of the *Decorator* pattern.

```
abstract class Component {
    abstract public String produce();
}

         class Decorator                      {



}

class Base extends Component {
    private String s;

    public Base(String s) {
        this.s = s;
    }

    @Override
    public String produce() {
        return s;
    }
}
```

19. Which of the following identifiers can we declare `x` so that the Java code below will compile successfully? (Circle all that apply)

    - `MyClass`
    - `Long`
    - `Object`
    - `Car`
    - `Plane`

```
public class MyClass implements CanMove {
    ____ x = new Plane();
}

interface CanMove {}

class Car extends MyClass {}

class Plane extends MyClass {}
```

20. Consider the following definition of the method `functionA`. If you know that the cyclomatic complexity of `functionB` is M then how would you find the cyclomatic complexity of `functionA`? Also, draw the control flow graph of `functionA` treating `functionB` as a single sequential block.

```
public static void functionA(Object o) {
    if (o == null) {
        return;
    }

    try {
        functionB();
    } catch (Exception e) {
        //
    }
}
```

21. What is the *Lack of Cohesion on Methods* (LCOM) of a class that has 4 methods and *no* instance variables? Show your work.

22. How do you add a dependency to a maven project? What information about the dependency do you need?

23. What are the arguments of the `doGet` and `doPost` methods of a servlet? What are three non-trivial things you can do with these arguments?

24. What design pattern is illustrated by servlets? This pattern is also used extensively in *frameworks*?

25. Suppose you want to log the result of an AJAX request in the Javascript console. What is the fundamental problem with the following code. (The dollar sign is the jQuery object.)

```
var res = "...";
$.ajax({
  url : "/myapplication/myendpoint",
  success : function(r) {
    res = r;
  }
});
console.log(res);
```

26. Consider the Jax-RS classes shown in the code-listing below. Assume the web application project is called `myproject` and that the name of your server is `hostname`. Answer the following.

    (a) At what URL would a client's request be handled by the `getMessage()` method?

(b) Show an example of a URL where the client's request would be handled by `testMethod()` and where the response would contain the URL parameter entered by the client?

```
//////// JAX-RS Service
@Path("/items")
public class MyService {

    @GET
    public Response getMessage() {
        return Response.ok("Hello, world.").build();
    }

    @GET
    @Path("{myName : [a-zA-Z]+}")
    public Response testMethod(@PathParam("myName") String myName,
                               @QueryParam("entered") String search) {
        return Response.ok(myName + " " + search).build();
    }
}

/////// JAX-RS Application
@ApplicationPath("/services")
public class MyApplication extends Application {
    private Set<Object> singletons = new HashSet<Object>();
    private Set<Class<?>> empty = new HashSet<Class<?>>();

    /* boilerplate code */
    /* assume MyService is registered properly */
}
```