1 DESIGN

1.1 Design Steps

- ~~What are the different steps in the Design Phase?~~
- ~~Describe the first three steps in the design phase.~~
- ~~What is abstraction leak? Use example of matrix multiplication to explain.~~
- ~~What are crosscutting concerns? Use an example of "performance concern" to explain.~~

1.2 Modular Design

- What is modular design?
- What does it mean when we say that a module is "cohesive"?
- What does it mean when we say that modules are "loosely coupled"?
- What are the three MAJOR benefits of modular design (describe these)?
- Why is it important to clearly specify interfaces of modules during design?
- Layering is a special case of modular design. What is unique or special about layered design? (i.e. what is layered design)?
- What is a KEY benefit of layered design? Give a real-world example of layered design.

1.3 Class Diagram

Draw a class diagram that captures ALL of the information in the below description.

A BANK has up to 1000 customers. It also has four loan managers.

A customer can have one or more of three different types of accounts: loan accounts, savings accounts, and checking accounts. There are three different types of loan accounts namely personal loans, car loans, and home-equity loans. Loan managers manage loan accounts. Accounts have attributes amount, interest, and time-period. An Account belongs to only one customer.

1.4 Sequence Diagram

Given the code fragment for a class below.

Assume that the **method1** method is being invoked for a particular **SomeClass object**. Draw a sequence diagram (NEATLY please).

```
class SomeClass {
  ArrayList<SomeType> someList;
  …
  double method2() { double  num = method3();  return num; }

  double method1 () {
        double sum = 0;
    for (SomeType s: someList) {
                  sum += s.someMethod();
                  sum += method2();
```

```
        }
    return sum;
  } // end of method2
} // end of SomeClass
```

1.5 Integration

- draw the V-model of development
- what are the different types of testing? When an error is found during testing what does it typically say about where the error originated – and what implications does it have on the costs to fix it?
- describe the big-bang integration strategy. what are the problems with this approach?
- describe the bottom-up integration strategy. what are the problems with this approach?
- describe the top-down integration strategy. why is this approach better than the other approaches?
- what are stubs?
- ~~describe the sandwich integration strategy.~~

1.6 Design Patterns

Facade, Observer, and MVC design pattern

- Clearly explain the problem that this pattern attempts to solve.
- Enumerate the benefits of using this pattern.
- Draw a class diagram for the pattern.
- Describe each of the participant classes and relevant methods/events/members of each class.
- Take ANY concrete example that uses the pattern (You can make up your own example). Name classes in this concrete example that map to the classes in the class diagram that you drew earlier).
- c) (10 points) Clearly explain the problem that this pattern attempts to solve and the benefits of using this approach over other approaches.

2 TESTING

2.1 When to stop Testing

Suppose you are a product manager, testing has been going on for the last two months, and you have to decide if the product is ready to be released. Describe THREE approaches to help you decide that **enough testing has been done** and that the product is ready to be released.

2.2 How to Automate

Broadly, the different steps in creating test code are:

1. Generate test cases (i.e. come up with test inputs and outputs)
2. Create a driver to run tests
3. Compare actual outputs with expected outputs

Discuss a) how to automate each step and b) how easy/hard it is to **automate each of these steps** (an example step is generating test inputs).

2.3 Importance of Automation

Why is it important to automate testing as much as possible?

2.4 Black Box Test Case generation

Describe three black box techniques to generate test cases (i.e. come up with test cases)?

2.5 Coverage

Describe two code coverage techniques.

2.6 For each step listed below, discuss a) how to automate each step and b) how easy/hard it is to automate each of these steps. Be sure to give an example.

Step 1: Generate test cases (i.e. come up with test inputs and expected outputs)

Step 2: Create a driver to run tests

Step 3: Compare actual outputs with expected outputs

3 PROCESS MODEL

3.1 (10 points) Processes

What are **five benefits** of having written down processes?

3.2 (10 points) Choosing a process model

How would one **decide which process model to use** for a particular project?

3.3 (10 pts) Process Models

Describe the Scrum process model. (Describe the roles, the ceremonies, and the artifacts).

3.4 Scrum Process Model

3.4.1 scrum basics

- what is a sprint?
- * what is a typical duration of a sprint?
- * what is done during a sprint?
- * what is not allowed during a sprint and why?

3.4.2 roles

- what is the difference between the product owner and the scrum master roles?
- how many persons make up a scrum team?

3.4.3 artifacts

- what is difference between product backlog and sprint backlog
- what does a sprint backlog represent?
- what is the purpose of a scrum board?
- what are the different items displayed on a scrum board?

3.4.4 ceremonies

- what is the goal of sprint planning?
- * how often is standup meeting held? when?
- * what does each developer do during a standup meeting?
- who participates in a sprint retrospective? what is done?
- who participates in a sprint review? what is done during the review?
-

3.5 Describe the SCRUM process model. Also, describe some shortcomings of the SCRUM process model (10)