

Total Points: 35

Answer ALL Questions

Question 1 (5 pts)

Please answer in the space provided. These are short answer questions.

1. Name the necessary conditions for perfect testing.

The two conditions for perfect testing are:

1. If all of our tests pass, then the program is correct, this can only be true if we test all the points in the input domain
2. We can tell whether the program will eventually halt and give a result from any t in our test set T . This relies on the solution to the halting problem, which has no solution.

2. How do composite diagrams help class diagrams in UML?

A composite diagram gives internals of a class or component, and can describe class relationships within a given context.

3. What is the difference between white box and black box testing?

Black box testing refers to testing where there is not access to the source code. The tests are derived entirely from specifications.

White box testing allows access to the source code. The internal structure of the program is used to derive tests.

4. What is mutation testing? Provide a definition.

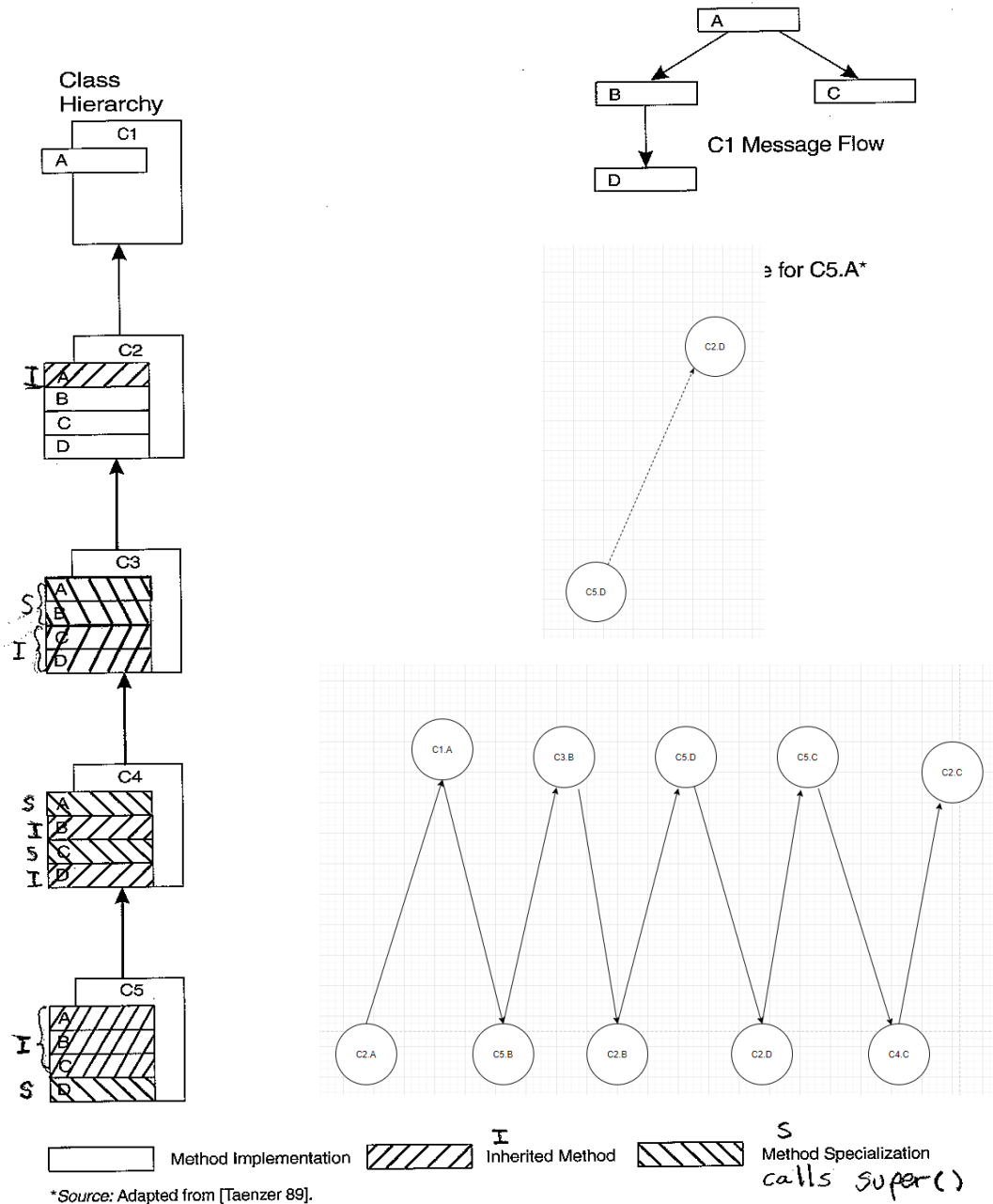
Mutation testing is making small changes to original code to create "mutants." These small changes are intentional faults and then test data is run through the mutants. A mutant is "killed off" if it produces different results than the original program.

5. Any path from s to t (where s is the starting node and t is the ending node) in a control flow graph G represents a potential execution path. Explain why some paths from s to t are infeasible.

There are some paths in which conditional statements are mutually exclusive. For example, `if(x>10){if(x<10){ //do something}}` The input x cannot be both greater and less than 10, so the code contained within second condition won't be reached.

Question 2 (10 pts)

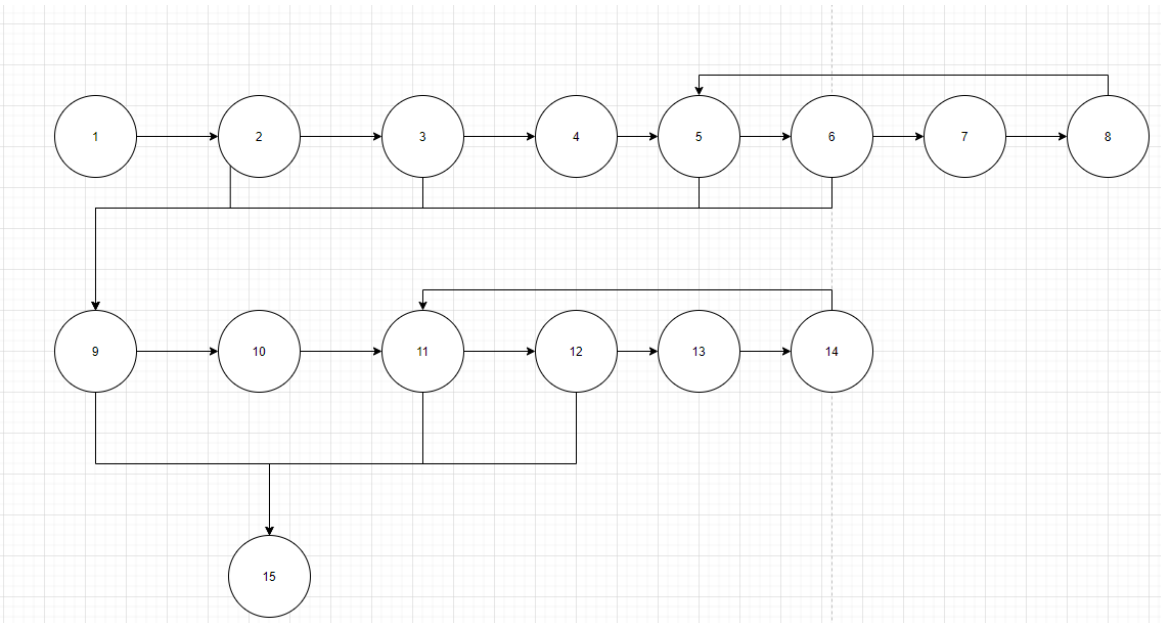
Draw the execution of the calls that exhibit the YoYo problem for a runtime trace of C5.D, and for C2.A (Draw both on the diagram below. Use a **dashed line** for C5.D and a **solid line** for C2.A).



Question 3 (10 pts)

Provide a Control Flow Graph (CFG) for the following code:

```
public int displayLastMsg( int nToPrint ) {  
    int np = 0;  
    if ( ( msgCounter > 0 ) && ( nToPrint > 0 ) ) {  
        for ( int j = lastMsg; ( j != 0 ) && ( np < nToPrint ); --j ) {  
            System.out.println( messageBuffer[j] );  
            ++np;  
        }  
        if ( np < nToPrint ) {  
            for ( int j = SIZE; ( j != 0 ) && ( np < nToPrint ); --j ) {  
                System.out.println( messageBuffer[j] );  
                ++np;  
            }  
        }  
    }  
    return np;  
}
```

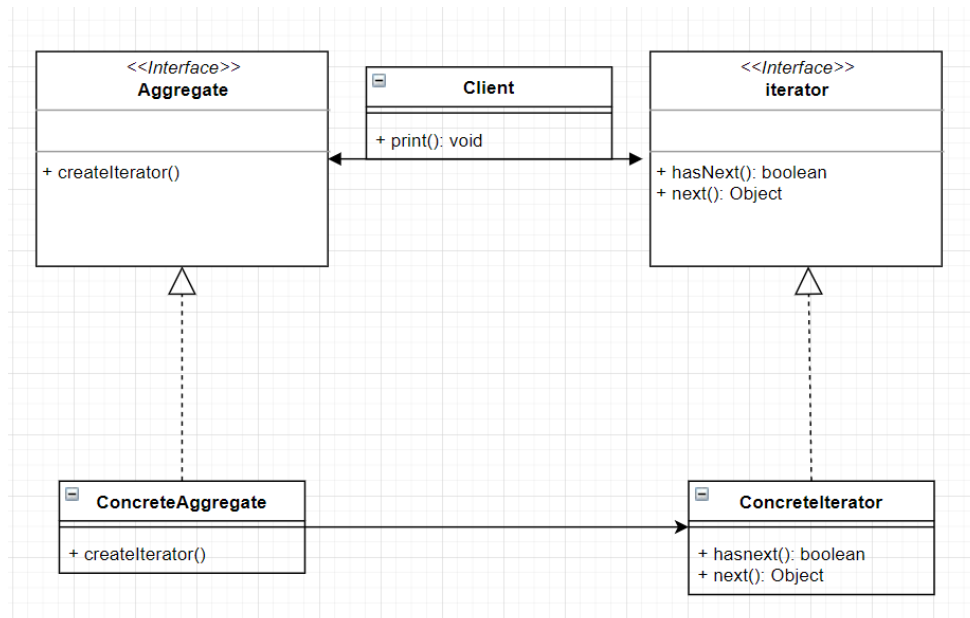


Question 4 (10 pts)

With respect to an iterator pattern, answer the following questions:

- a) What are the advantages of using an Iterator pattern? (1 pt)
- b) Draw the UML class diagram for an Iterator pattern (1 pt)
- c) Assume the container that implements the Iterator pattern stores integers (in any order). Write **client code** that uses the Iterator to traverse through all the integers (4 pts)
- d) Write the **next()** method of the Iterator so that only every third number of the contained integers is returned. (4 pts)

- a) Iterators can be used to access the elements of an aggregate object without exposing its underlying representations.
- b)



- c)

```
class Client{
    ConcreteAggregate[] aggregateList;
    public Client(ConcreteAggregate aggregateList){
        this.aggregateList = aggregateList;}
    public void print(){
        Iterator iterator = aggregateList.createIterator();
        while(iterator.hasNext()){
            int a = iterator.next();
            system.out.println(a);}}
```
- d)

```
public int next() {
    int number = integerList[pos];
    pos += 3;
    return number;
}
```