SYSC 4101

Laboratory 9

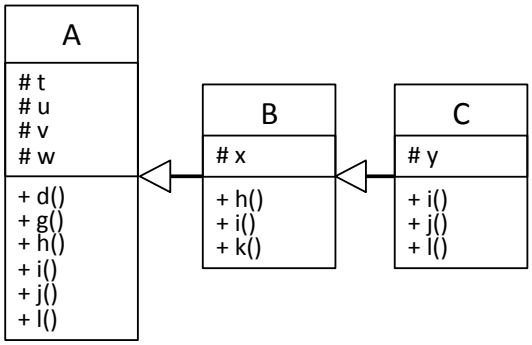
Exercise A

Consider the simple inheritance hierarchy below: class `C` is a subclass of class `B`; class `B` is a subclass of class `A`.
The table on the right indicates where methods are defined, or re-defined. It also indicates what calls each method in the hierarchy is making. For instance:

- Method `d()` (row 1) defined in class `A` calls method `g()`;
- `A.i()` (row 8) indicates that method `i()`, which is re-defined in class `B` calls the version of `i()` that is defined in parent class `A`.

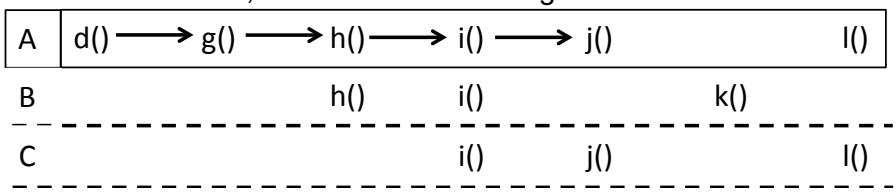
The last two columns indicate which attributes are defined and used. For instance:

- Method `j()` of `A` (row 5) defines attribute `v` of `A` and uses attribute `w` of `A`;
- Method `l()` of `C` uses attributes `v` inherited from `A` (row 12).

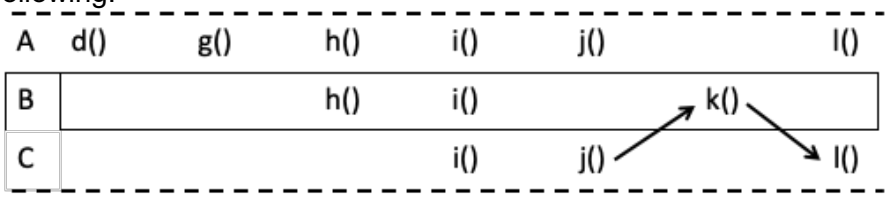


	Class	Method	Calls	Definitions	Uses
1	A	d()	g()		
2		g()	h()		
3		h()	i()	A.u, A.w	
4		i()	j()		A.u
5		j()		A.v	A.w
6		l()			A.v
7	B	h()	i()	B.x	
8		i()	A.i()		B.x
9		k()	l()		
10	C	i()	B.i()	C.y	
11		j()	k()		C.y
12		l()			A.v

Consider a call to method `d()` on an instance of class `A`. The sequence of method calls is illustrated by the following figure: d() calls g() (as per the table), which calls h(), which calls i(), which calls j() and all those calls happen on the instance of class A. Since all the calls happen on methods defined by the instance of class A, all the arrows in the figure below are on the row for A.

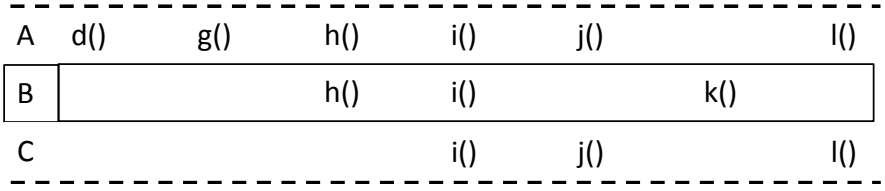


If instead you consider a call to method j() on an instance of class C, the sequence of calls is the following:



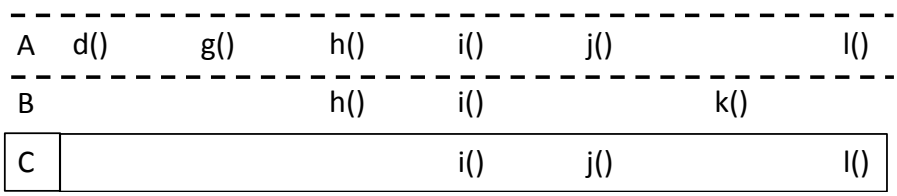
Question 1. [5 marks]

What is the sequence of method calls when `d()` is executed on an instance of class `B`?
Fill the following diagram with arrows between methods to illustrate the sequence of method calls. You can find a PowerPoint file online to help you do the drawing. You only need to place arrows correctly.
In what order are attribute values defined and used? Is there any problem?



Question 2. [5 marks]

Similarly to the question above, what is the sequence of method calls when `d()` is executed on an instance of class `C`? Fill the following diagram.
In what order are attribute values defined and used? Is there any problem?



Question 3. [5 marks]

According to the Hierarchical Incremental Testing principle, and assuming that all the methods defined by `A` have been properly tested, do methods `d()`, `g()`, `j()` and `l()` need to be re-tested in the context of class `B`? Justify your answer.

Question 4. [5 marks]

Class `B` provides a new implementation for, i.e., overrides, methods `h()`, and `i()`. These new implementations need to be tested. What can you say about the test scaffolding that has been used to test the implementation of these methods as defined in class `A`? Justify your answer.

Question 5. [5 marks]

According to the Hierarchical incremental Testing principle, and assuming that class `B` has been properly tested after class `A`, which methods of class `C` need to be tested or re-tested and which do not need retesting? Justify your answer.

Exercise B

Suppose you are dealing with a piece of software that receives a stream of integer values, for instance from a network feed, and is tasked with computing the factorial of each received integer value.
The code that performs the computation is the following:

```
public class Factorial {
	public static void compute(FeedOfIntValues f) {
		int i;
		while (f.hasNext()) {
			i = f.getNextIntValue();
			System.out.println("Factorial of "+i+" = "+factorial(i));
		}
	}
	private static long factorial(int num) { //Assumes num > 0
		if (num>=1)
			return num * factorial(num-1);
		else
			return 1;
	}
}
```

The integer values to consider are collected from a "feed": parameter `f` of type `FeedOfIntValues` in the code above. For convenient design, since the factorial computation does not need to know where the integer values are coming from, `FeedOfIntValues` is an interface:

```
	public interface FeedOfIntValues {
		public boolean hasNext();
		public int getNextIntValue();
	}
```

The code above is available on the course web site.

Question 1 (5 pts)

Identify input data as well as the corresponding expected outputs, for testing function `compute()`, while satisfying the all-edges coverage criterion of the methods of class `Factorial`.

Question 2 (10 pts)

You now have to implement those test cases. However, since the integer value feed does not yet exist, you need to stub it. Create stubs for your tests (they should be concrete implementations of the interface), compile and execute your tests. Note that you may likely need more than one stub so you keep the design and implementation of the stubs as simple as possible.