Practice Problems 2.0

1. Answer the following questions for classes `ClassA` and `ClassB`.
   a. Which method overrides a method in superclass?
   b. Which method hides a method in superclass?

```
class ClassA {
    public void methodOne(int i) {
    }
    public void methodTwo(int i) {
    }
    public static void methodThree(int i) {
    }
    public static void methodFour(int i) {
    }
}

class ClassB extends ClassA {
    public static void methodOne(int i) {
    }
    public void methodTwo(int i) {
    }
    public void methodThree(int i) {
    }
    public static void methodFour(int i) {
    }
}
```

2. Write the output of the following code when the main method of class `QuestionTwoChecker` `is executed.`
   You have to explain each output after each println statement, e.g.,

   First println statement: b
   Reason: `System.out.println(elements[i]);`
   It means `toString` method of **`class`** `A` is executed. Though **`class`** `A` does not have a default `toString method,` it inherits a `toString` method from `class B` which prints `b.`
   Hence, the `println` statement prints `b.`

   Without explanation no answer will be taken into consideration for marking.

```
class C {
    public String toString() {
        return "c";
    }

    public void method1() {
        System.out.println("c 1");
    }

    public void method2() {
        System.out.println("c 2");
    }
}

class B extends C {
    public String toString() {
        return "b";
    }

    public void method2() {
        System.out.println("b 2");
    }
}

class A extends B {
    public void method2() {
        System.out.println("a 2");
    }
}

class D extends B {
    public void method1() {
        System.out.println("d 1");
    }
}

public class QuestionTwoChecker{
    public static void main(String[] args) {
        C[] elements = {new A(),
                new B(),
                new C(),
                new D()};
        for (int i = 0; i < elements.length; i++) {
            System.out.println(elements[i]);
            elements[i].method1();
            elements[i].method2();
            System.out.println();
        }
    }
}
```

3. Implement the missing classes shown in the diagram shown below.

   For each class, the method signatures are provided, e.g., class `Ham` should have 3 methods named `a(), b(), toString()` which prints `Ham a, Ham b and Ham` respectively.

   NB: In class `Lamb` method `a()` prints `Ham a`. To implement this you **cannot write** `System.out.println("Ham a").` You have to use the concept of **inheritance** in Java. Same thing goes for the similar methods in the diagram.

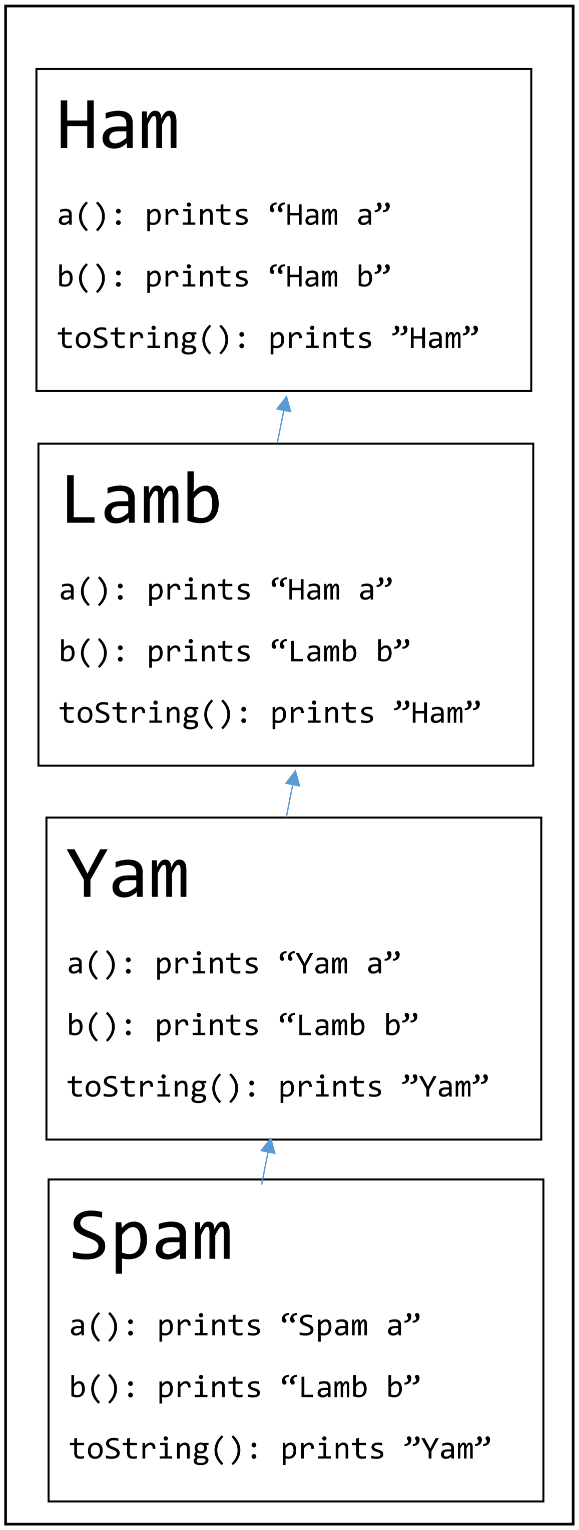
   A tester class `Polymorphism` is provided to test the classes with output in the next page.

```java
public class Polymorphism {
    public static void main (String [] args){
        Ham[] food = { new Spam(), new Yam(),
                       new Ham(), new Lamb() };

       for (int i = 0; i < food.length; i++) {
           System.out.println(food[i]);
           food[i].a();
           food[i].b();
           System.out.println();
       }
    }
}
```

Output:

```
Yam
Spam a
Lamb b
Yam
Yam a
Lamb b
Ham
Ham a
Ham b
Ham
Ham a
Lamb b
```


Ham
a(): prints “Ham a”
b(): prints “Ham b”
toString(): prints ”Ham”
Lamb
a(): prints “Ham a”
b(): prints “Lamb b”
toString(): prints ”Ham”
Yam
a(): prints “Yam a”
b(): prints “Lamb b”
toString(): prints ”Yam”
Spam
a(): prints “Spam a”
b(): prints “Lamb b”
toString(): prints ”Yam”