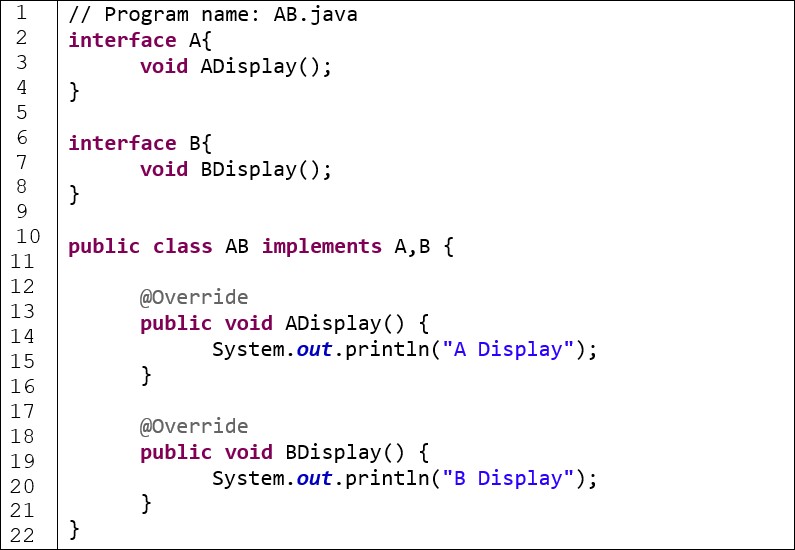
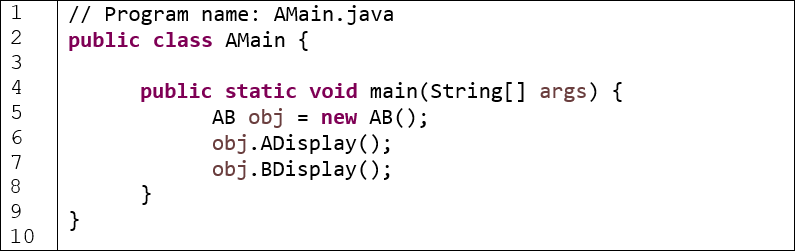
# Section A:

*Interface Concept*

**TK1143 - Tutorial 4 Polymorphism Using Interface**

1. What is the output of the codes below?



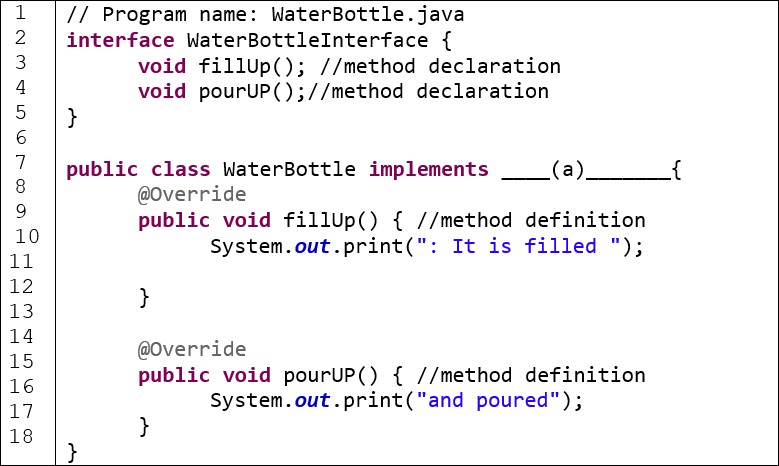


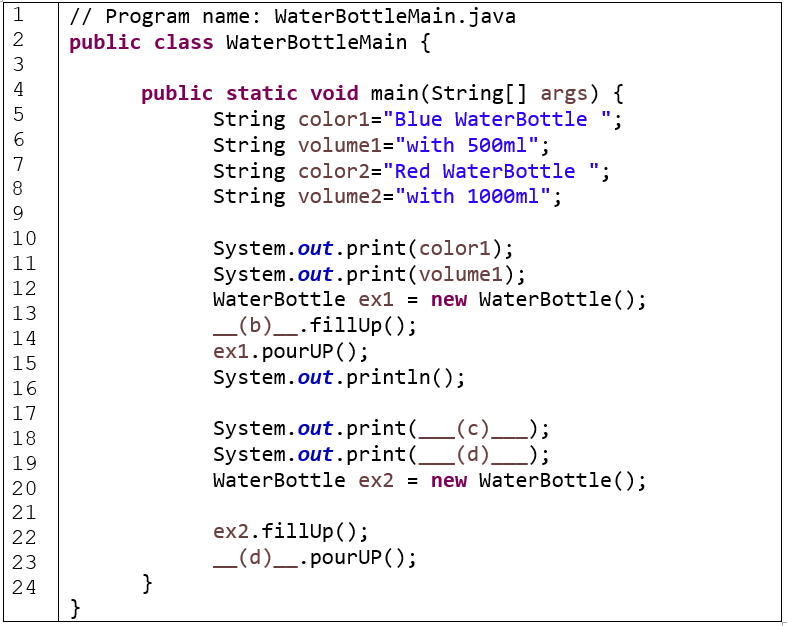
Answer:

A Display

B Display

1. Answer the following questions based on the code below:
2. Fill in the blank of the codes:





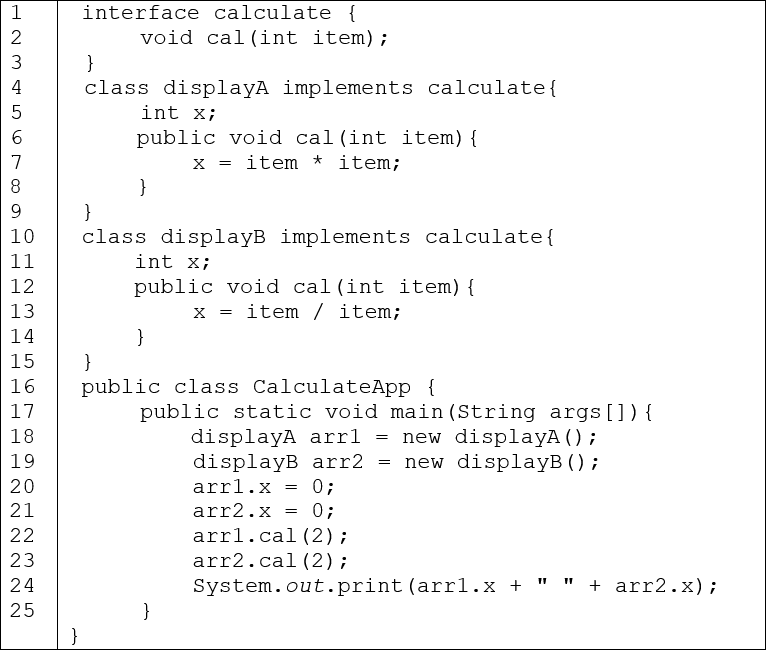
Answer:

1. WaterBottleInterface
2. ex1
3. color2
4. volume2 , ex2
5. What is the output of the code above? Answer:

Blue WaterBottle with 500ml: It is filledand poured

Red WaterBottle with 1000ml: It is filledand poured

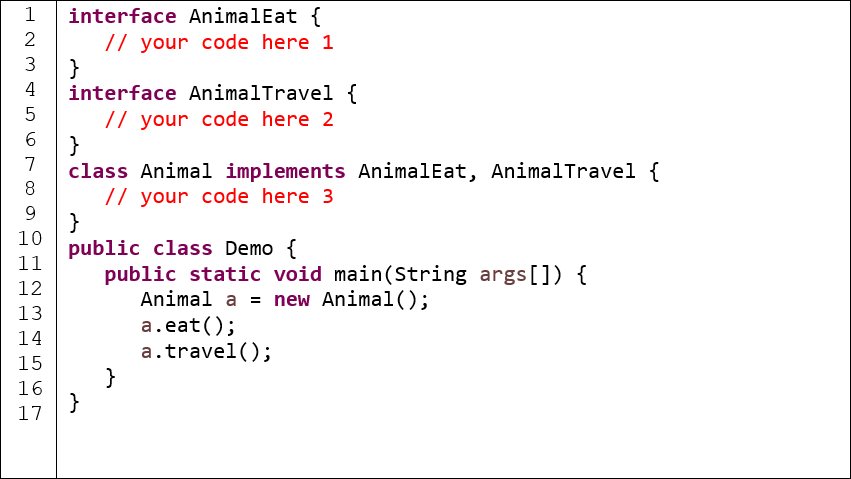
1. What is the output of the following code?



Answer:

4 1

1. Write the codes to achieve the output below?



**Output:**

Animal is eating Animal is travelling

Answer:

void eat();

void travel();

public void eat(){

System.out.println(“Animal is eating”);

}

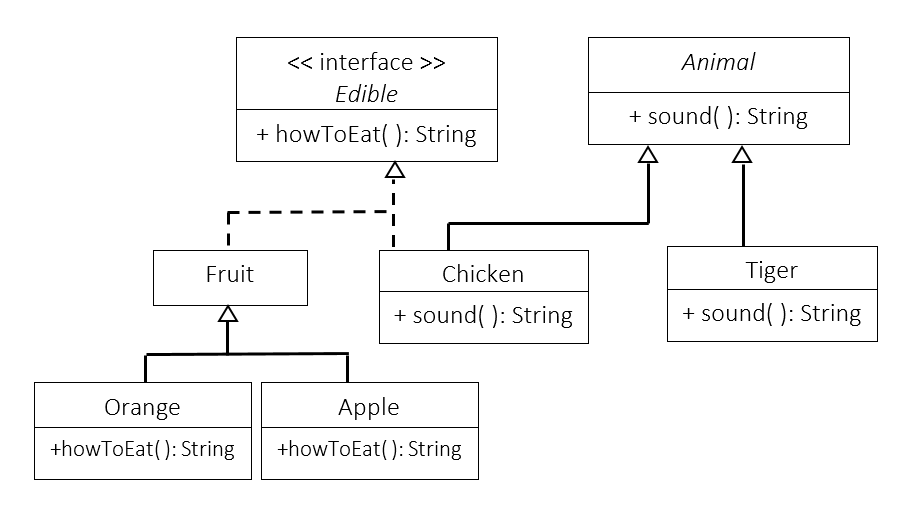
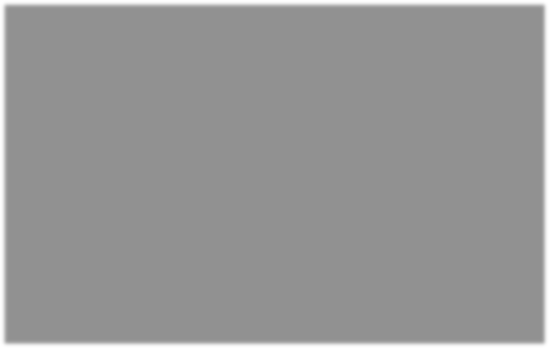
Public void travel(){

System.out.println(“Animal is Travelling”);

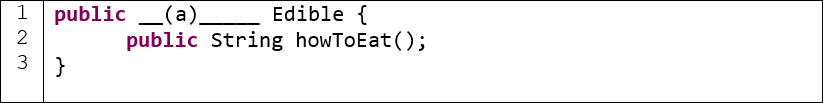
}

*Polymorphism Using Interface*

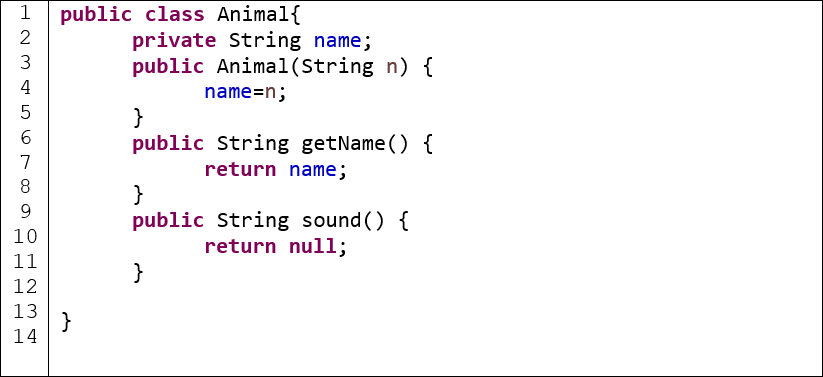
Based on the diagram below, answer the questions follow:



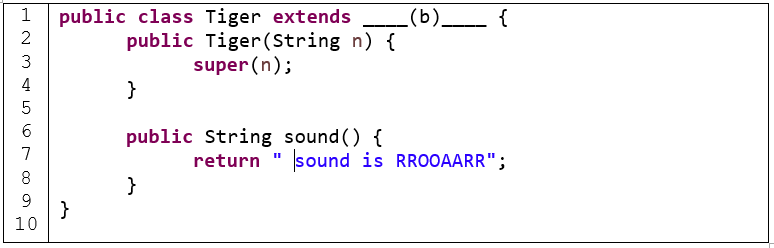
1. Fill in the blanks:
2. Interface **Edible** (*Edible.java*)



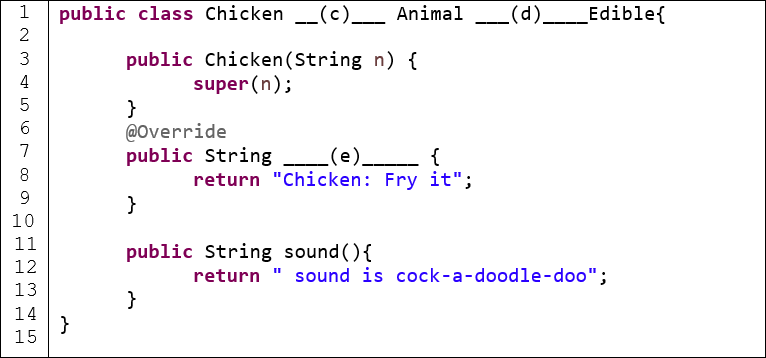
1. Class **Animal** (*Animal.java*)



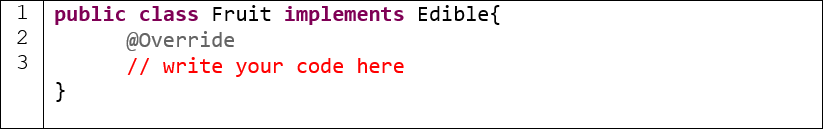
1. Class **Tiger** (*Tiger.java*)



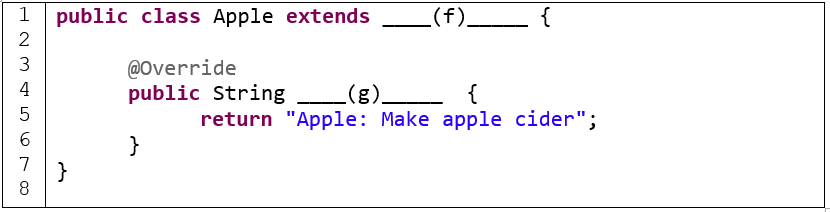
1. Class **Chicken** (*Chicken.java)*



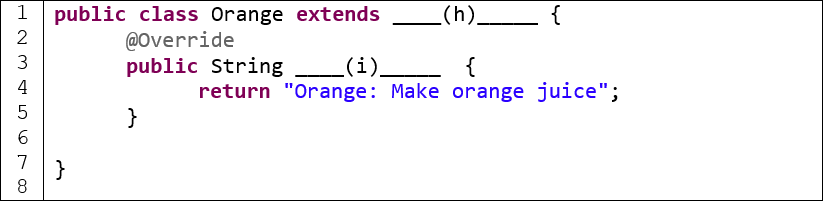
1. Class **Fruit** (*Fruit.java*)



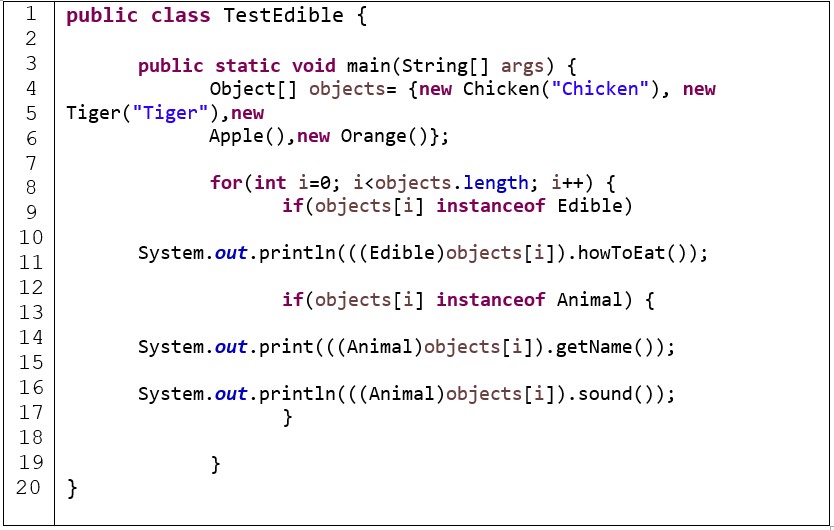
1. Class **Apple** (*Apple.java*)



1. Class **Orange** (*Orange.java*)



1. Class Application (*TestEdible.java*)



Answer:

1. interface
2. Animal
3. extends
4. implements
5. howToEat()
6. Fruit
7. howToEat()
8. Fruit
9. howToEat()
10. Write the override method in class Fruit that implements interface Edible. Answer:

public String howToEat(){

return null;

}

1. What is the output for the codes above? Answer:

Chicken: Fry it

Chicken sound id cock-a-doodle-do

Tiger sound is RROOAARR

Apple: Make apple cider

Orange: Make orange juice