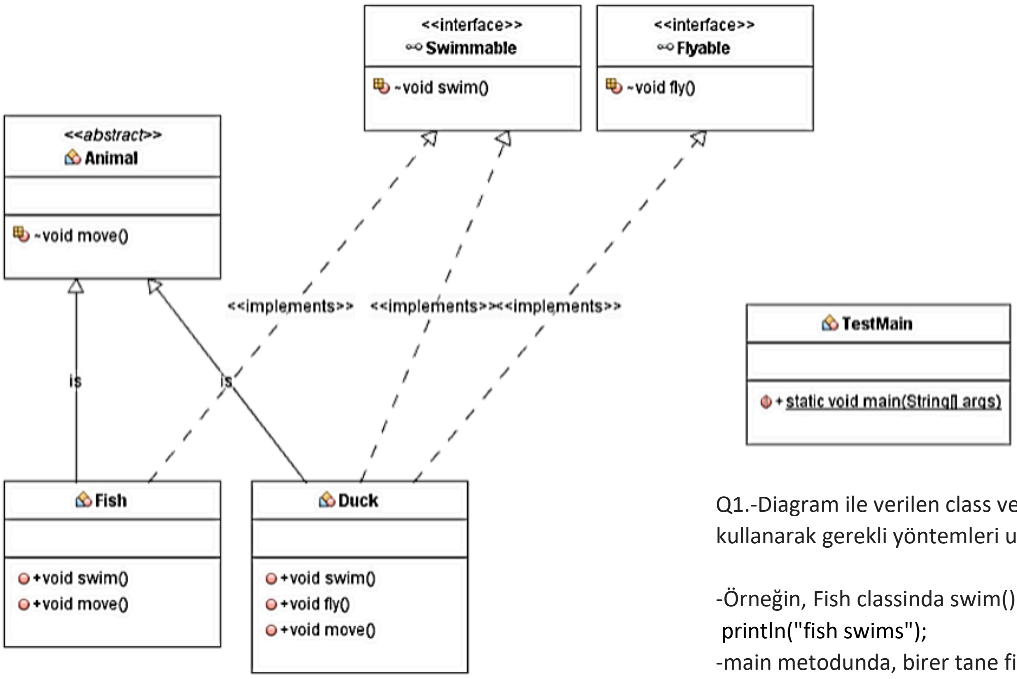
NAME:
ID:

	Q1	Q2	Q3	Q4	Q5	TOTAL
SCORE						

OBJECT ORIENTED PROGRAMMING FINAL EXAM

Q1 (25p): Write the classes and interfaces given by following diagram. In the main method, create fish and dog objects. Implement the necessary methods using println(). For instance swim() in class Fish: println("fish swims");



Q1.-Diagram ile verilen class ve interfaceleri yazınız. println() kullanarak gerekli yöntemleri uygulayın.

-Örneğin, Fish classinda swim():
println("fish swims");
-main metodunda, birer tane fish ve dog nesneleri oluşturun.

Q2(28p): Using the relationship given in the previous question, check if the following compiles and runs.

```
public class TestMain {
    public static void main(String[] args){
      Animal a= new Duck();
      a.swim();
      a.fly();
      a.move();
      Flyable f =(Flyable)new Fish();
      Animal a = new Duck();
      a.fly();
   }
}
```

//WRITE YOUR ANSWER IN THE TABLE

Statement	Compile?	Run?	If compiles or runs, explain why? If not, Correct it (if it is possible)
`Animal a= new Duck();`			
`a.swim();`			
`a.fly();`			
`a.move();`			
`Flyable f = (Flyable)new Fish();`			
`f = new Duck();`			
`f.fly();`			

Q2.Önceki soruda verilen ilişkiyi kullanarak, tabloda verilen ifadelerin derlenip calistigini kontrol ediniz. Tam puan almak icin cevabinizi aciklayin.

Q3 (18p): 1.What is multiple inheritance in Java? Write the differences between classes and abstract classes.
[Java'da çoklu miras nedir? Classlar ve abstract classlar arasındaki farkları yazın.] // Write answer here

Q4 (24): Write the output. <u>Explain your answer. Otherwise you will get half score.</u>

```
abstract class Worker {
private String name;
protected int hour;
protected int wage;
    public Worker(String name, int hour) {
        this.name = name;
        this.hour = hour;
    }

    public Worker(String name){
        this.name = name;
    }
    abstract int Salary();
    @Override
    public String toString() {
        return name +" "+ this.Salary();
    }

    int baseSalary(){
        return 1000;
    }
}

class Engineer extends Worker{
    public Engineer(String name) {
        super(name);
        this.hour =10;
        this.wage = 50;
    }

    @Override
    int Salary() {
        return this.hour * wage +
super.baseSalary();
    }
}

class ChiefEngineer extends Worker{
   public ChiefEngineer(String name, int hour) {
        super(name, hour);
        this.wage = 20;
    }

    @Override
    int Salary() {
        return this.hour * wage * 2 +
super.baseSalary();
    }
}

class TestClass{
    public static void main(String[] args){
      Worker w1 = new Engineer("newbie");
      Worker w2 = new ChiefEngineer("senior", 40);

      System.out.println(w1);
      System.out.println(w2);
    }
}
```

Q4.Write answer in here

Q4. Çıktıyı yazın<u>. Cevabınızı açıklayın. Aksi takdirde yarım puan alırsınız.</u>

YOU HAVE 90 MINS.

ANSWER QUESTION 2, 3, 4 in the given spaces.
<u>(Soru 2,3,4 uzerinde cevaplanacaktir)</u>

GOOD LUCK!

Q5:
Book adında bir class yazın. Classin 2 attribute'u olmalı: name (String) ve price (double).
a) Constructor bu 2 attribute atamak zorundadır.
b) Toplam fiyatı döndüren bir totalPrice () metodu oluşturun (price +% 8 * price).
c) Bu Classtan 4 (b1,b2,b3,b4) nesne oluşturun ve hepsini books olarak adlandırılan LinkedList'e ekleyin. d) Listedeki nesneleri toplam fiyata göre sıralayın. (İpucu: Comparable interface'ini kullanın)
e) Listede sıralanmış olan nesneleri Iterator kullanarak yazdırın.

Q5(25p): Write a class called Book. The class must have 2 attributes: name(String) and price(double).

a) Class constructor will have to set these 2 attributes.
b) Create a totalPrice() method, which returns total price (price + %8 * price).
c) Create 4 objects (b1, b2, b3,b4) from this class and add them all to a LinkedList called **books**.
d) <u>Sort objects in the list by total Price</u> **(Hint: Use Comparable interface)**
e) Print sorted objects in the list using **Iterator**.