

Full Name:
Id:
Program: Day class <input type="checkbox"/> Night class <input type="checkbox"/>

SCORE	Q1	Q2	Q3	Q4	TOTAL

UNIKA | COMPUTER ENGINEERING DEPARTMENT
Object Oriented Programming | Midterm | Fall | 21.11.2022 | 13.00 | Duration: 80 mins

<pre>// All the classes in the same file under Company.java name 1. package midexam2023; 2. enum Position{ 3. PROGRAM(100), 4. CONSULTANT(200), 5. DESIGNER(300); 6. private final int id; 7. public int getId(){ 8. return id; 9. } 10. } 11. public class Employee{ 12. private int empId; 13. private String name; 14. public static final String companyName="Google"; 15. public Employee(String name, int empId,String cName){ 16. this.name=name; 17. this.empId= empId; 18. this.companyName= cName; 19. } 20. public String getName(){ 21. return this.name; 22. } 23. public String getId(){ 24. return this.empId; 25. }</pre>	<pre>26. static void printName(){ 27. System.out.println(" Name:"+name); 28. } 29. @Override 30. public String toString(){ 31. return this.empId+": name:"+this.name; 32. } 33. }//end of employee class 34. public class Company { 35. public void main(String[] args) { 36. Employee employee1= new Employee("Ahmed", 100, "Google"); 37. System.out.println(employee1); 38. employee1.name="Lana"; 39. Employee employee2= new Employee("Omer", 200, "MS"); 40. System.out.println(employee2.getName()); 41. Employee.getName(); 42. Employee employee3= new Employee("Ali", 300.5f, "Meta"); 43. System.out.println(employee4); 44. } 45. }</pre> <p>QUESTION.1 [20p]: Find 10 errors in the following Java code. The errors are both compiler and logical errors. For each error, specify the line number and briefly explain how to fix it.</p> <p>SORU-1 Verilen Java kodunda 10 hata bulun. Hatalar hem derleyici hem de mantıksal hatalardır. Her hata için satır numarasını belirtin ve nasıl düzeltileceğini kısaca açıklayın.</p>
---	--

QUESTION 2 AND 3 ARE CONNECTED! | 2. VE 3. SORULAR BAĞLANTILIDIR!

<p>Car</p> <pre>- brand : String - maxSpeed : int - year : int ~ Car(year : int) ~ Car(year : int, brand : String) + Car(year : int, maxSpeed : int, brand : String) + setBrand(brand : String) + setYear(year : int) + setMaxSpeed(maxSpeed : int) + getMaxSpeed() : int + getYear() : int + getBrand() : String + getCarInfo() : String</pre>	<p>Factory</p> <pre>~ cars : Car[] ~ Factory(cars : Car[]) ~ printAllCarsInfo()</pre> <p>QUESTION 2[30pt]: Write necessary classes for given UML diagram. All the classes are declared in the same package. Don't repeat the codes in the constructors and do not violate data security!</p> <p>SORU 2.Verilen UML diyagramı için gerekli sınıfları yazınız. Tüm sınıflar aynı pakette verilmiştir. Yapıcılarda (constructor) kodları tekrarlamayın ve veri güvenliğini ihlal etmeyin!</p>
---	---

QUESTION.3[26pt]: THIS QUESTION IS RELATED TO QUESTION 2.

Considering UML in Question 2, create three car objects, each using a different constructor in the main method. Also, create a factory object. Call the printAllCarsInfo() method which prints all the cars in the factory. getCarInfo() int Car class method should return attributes of a car.

In setYear method, year argument/parameter cannot be bigger than 2022. If it is, throw an exception with a message. You should handle the exception in the main method.

SORU 3. BU SORU 2. SORU İLE BAGLANTILIDIR,

2. Soruda verilen UML'I göz önünde bulundurarak, her biri için farklı kurucu (constructor) kullanarak üç car nesnesini main metodunda oluşturun.

Ayrıca, bir tane de factory nesnesi oluşturun. Factory'deki tüm car nesnelerini ekrana yazdıran printAllCarsInfo() metodunu çağırın. Car classındaki getCarInfo() car nesnesinin tüm özelliklerini döndürmelidir.

setYear metodunda year argümanı/parametresi 2022'den büyük olamaz. Eğer olursa, bir mesajla istisna (exception) fırlatılmak zorundadır. İstisnayı main metodda ele almalısınız (handle).

QUESTION 4. [24p]: Write outputs in the given table. All the classes are declared in the same package.

Çıktıları verilen tabloya yazın. Tüm sınıflar aynı pakette yazılmıştır.

<pre>enum AnimalType{ MAMMAL(100), BIRD(200), RAP TILE(300), FISH(400); private final int typeId; private AnimalType(int typeId) { this.typeId = typeId;} int getId(){ return typeId;} } public class Animal { private String name; private int id; static int counter; AnimalType animaltype; Animal(String name,AnimalType type){ this.name = name; this.animaltype = type; this.id= ++counter; } int getId(){ return id; } @Override public String toString(){ return name +" is a " + animaltype; } } // End of Animals class</pre>		<pre>public class Zoo { static int index; Animal[] animals; public Zoo(int i){ this.animals= new Animal[i]; } void addAnimal(Animal a){ animals[index] = a; index++; } void printAnimals(){ for(Animal a:animals) System.out.println(a); } } // End of Zoo class public class Exam2022 { public static void main(String[] args) { Animal a1= new Animal("Cat",AnimalType.MAMMAL); Animal a2= new Animal("Dove",AnimalType.BIRD); System.out.println(Animal.counter); //1 System.out.println(a1); //2 Animal a3= new Animal("Salmon",AnimalType.FISH); a2.animaltype=a3.animaltype; Animal a4= new Animal("Snake",AnimalType.RAP TILE); System.out.println(a3.getId()); //3 System.out.println(a2.animaltype.getId()); //4 Zoo myZoo= new Zoo(2); myZoo.addAnimal(a1); myZoo.addAnimal(a4); myZoo.printAnimals(); //5 } }</pre>	
(01)	2	(04)	400
(02)	Cat is a MAMMAL	(05)	Cat is a MAMMAL Snake is a RAP TILE
(03)	3		

Q2-Q3

```
public class Car {
    private String brands;
    private int maxSpeed;
    private int year;

    Car(String brands, int year) {
        this(brands, 180, year);
    }
    Car(String brands) {
        this(brands, 180, 2022);
    }

    public Car(String brands, int maxSpeed, int year) {
        setBrands(brands);
        setMaxSpeed(maxSpeed);
        setYear(year);
    }

    public void setMaxSpeed(int maxSpeed) {
        this.maxSpeed = maxSpeed;
    }

    public void setYear(int year) {
        if(year > 2022)
            throw new IllegalArgumentException("year cannot be greater");
        this.year = year;
    }

    public void setBrands(String brands) {
        this.brands = brands;
    }

    public int getYear() {
        return year;
    }

    public int getMaxSpeed() {
        return maxSpeed;
    }

    public String getBrands() {
        return brands;
    }
    public String printCarInfo() {
        return String.format("BRAND: %s YEAR: %d SPEED: %d",
            this.brands, this.year, this.maxSpeed);
    }
}

public class Factory {
    Car[] cars;

    Factory(Car[] cars) {
        this.cars = cars;
    }

    void printAllCarInfos() {
        for(Car car: cars)
            System.out.println(car.printCarInfo());
    }
}

public class Main {
    public static void main(String[] args) {
        try{
```

```

        Car car1 = new Car("BMW");
        Car car2 = new Car("XD", 2022);
        Car car3 = new Car("FERRARI", 350, 1985);
        Car[] cars = new Car[]{car1, car2, car3};
        Factory factory = new Factory(cars);
        factory.printAllCarInfos();
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
}

```

Q1.

```

enum Position{
PROGRAM(100),
CONSULTANT(200),
DESIGNER(300);
private final int id;

```

//1- Must have constructor

```

public int getId(){
return id;
}
}

```

//2- file name should have same name as the public class, also we can't have 2 public

```

public class Employee{
private int empld;
private String name;
public static final String companyName="Google";
public Employee(String name, int empld,String cName){
this.name=name;
this.empld= empld;
this.companyName= cName; //3- cannot assign a value to a final variable
}

```

```

public String getName(){
return this.name;
}

```

```

public String getId(){
return this.empld; //4- should have same return datatype
}

```

```

static void printName(){
//5- non static variable cannot be referenced from a static method
System.out.println(" Name:"+name);
}

```

@Override

```

public String toString(){
return this.empld+": name:"+this.name;
}
}

```

```

public class Company {
public static void main(String[] args) //6- main method should be static
Employee employee1= new Employee("Ahmed", 100, "Google");
System.out.println(employee1);
employee1.name="Lana"; // 7- name has a private access
Employee employee2= new Employee("Omer", 200, "Microsoft");
}
}

```

```
System.out.println(employee2.getName());
```

```
Employee.getName();//8- non static method cannot be referenced from a static method
```

```
Employee employee3= new Employee("Ali", 300.5f, "Meta");//9- incompatible type, passing float value to int
```

```
System.out.println(employee4);// 10- object not exist
```

```
}
```

```
}
```