

[Dashboard](#) / [My courses](#) / [ITB_IF2212_2_2223](#) / [Praktikum 4 Polymorphism](#) / [Praktikum 4 - Polymorphism \(K01 & K02\)](#)

Started on	Tuesday, 21 February 2023, 8:10 AM
-------------------	------------------------------------

State	Finished
--------------	----------

Completed on	Tuesday, 21 February 2023, 9:04 AM
---------------------	------------------------------------

Time taken	54 mins 37 secs
-------------------	-----------------

Grade	300.00 out of 300.00 (100%)
--------------	-----------------------------

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Polymorphism merupakan konsep dimana sebuah subclass dapat mendefinisikan sifatnya sendiri dan dapat menggunakan fungsionalitas yang sama dari parent classnya. Pada soal ini, akan diuji pemahaman kalian terkait polymorphism

Pada nomor ini, terdapat sebuah parent class yang dapat dilihat pada [Car.java](#), dimana:

- Merupakan parent class
- Memiliki konstruktor berupa numberOfWheels dan engineCapacity
- Memiliki getter untuk masing-masing atributnya
- Mengeluarkan suara "Ngenggg"
- Mengeluarkan klakson "Din"
- Dapat mendeskripsikan dirinya dengan format "Mobil ini memiliki a roda dengan kapasitas mesin b cc", dimana a adalah numberOfWheels dan b adalah engineCapacity

Praktikan akan diminta untuk melengkapi 2 buah file, yaitu Bus.java dan Perari.java, dimana:

1. Bus.java

- Merupakan subclass dari Car
- Memiliki konstruktor yang sama dengan parentnya, dengan tambahan berupa passengerCapacity dan isUsedForLongTrip
- Memiliki getter untuk masing-masing atributnya
- Mengeluarkan suara "Cesss"
- Mengeluarkan klakson "Notnot"
- Dapat mendeskripsikan dirinya sesuai dengan keadaan:
 1. Apabila isUsedForLongTrip true, maka mengembalikan: "Mobil ini memiliki a roda dengan kapasitas mesin b cc, memiliki kapasitas penumpang c orang, dan digunakan untuk perjalanan jauh", dimana a adalah numberOfWheels, b adalah engineCapacity dan c adalah passengerCapacity
 2. Apabila isUsedForLongTrip false, maka mengembalikan: "Mobil ini memiliki a roda dengan kapasitas mesin b cc, memiliki kapasitas penumpang c orang, dan **tidak** digunakan untuk perjalanan jauh", dimana a adalah numberOfWheels, b adalah engineCapacity dan c adalah passengerCapacity

2. Perari.java

- Merupakan subclass dari Car
- Memiliki konstruktor yang sama dengan parentnya, dengan tambahan berupa averageSpeed dan isInsurance
- Memiliki getter untuk masing-masing atributnya
- Mengeluarkan suara "Brmmm"
- Mengeluarkan klakson "Siuuu"
- Dapat mendeskripsikan dirinya sesuai dengan keadaan:
 1. Apabila isInsurance true, maka mengembalikan: "Mobil ini memiliki a roda dengan kapasitas mesin b cc, memiliki kecepatan rata-rata c km/h, dan sedang dalam perlindungan asuransi", dimana a adalah numberOfWheels, b adalah engineCapacity dan c adalah averageSpeed
 2. Apabila isInsurance false, maka mengembalikan: "Mobil ini memiliki a roda dengan kapasitas mesin b cc, memiliki kecepatan rata-rata c km/h, dan sedang **tidak** dalam perlindungan asuransi", dimana a adalah numberOfWheels, b adalah engineCapacity dan c adalah averageSpeed

Submit zip file(nama dibebaskan) yang berisi [Bus.java](#) dan [Perari.java](#)

Java 8

 [Car.zip](#)**Score: 100****Blackbox**

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	25	Accepted	0.08 sec, 31.00 MB
2	25	Accepted	0.08 sec, 27.87 MB
3	25	Accepted	0.08 sec, 29.07 MB
4	25	Accepted	0.08 sec, 28.43 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Polymorphism - CarPrinter

Buatlah kelas **CarPrinter** yang berguna untuk mencetak informasi **Car** dan kelas turunannya (**Perari** dan **Bus**) dari beberapa mobil. Menggunakan kelas yang telah dibuat sebelumnya, kelas **CarPrinter** memiliki atribut **testCar** bertipe **Car** dan **carList** yang berbentuk **ArrayList of Car**. **CarPrinter** juga memiliki metode metode sebagai berikut:

1. **addCar**, yang menerima parameter **newCar** dengan tipe **Car**, menambahkannya ke dalam atribut **carList**, dan tidak mengembalikan apapun (*Hint: Penambahan elemen ke dalam sebuah objek ArrayList dapat menggunakan method add(E element) yang sudah tersedia*)
2. **getCarList**, yang mengembalikan atribut **carList**
3. **getTestCar**, yang mengembalikan atribut **testCar**
4. **setTestCar**, yang menerima parameter **carType** bertipe **String**, kemudian menjalankan aksi berikut berdasarkan isi parameter **carType** dan tidak mengembalikan apapun:
 - Apabila **carType** bernilai **Car**, maka atribut **testCar** akan menjadi sebuah objek baru dengan kelas **Car** yang memiliki parameter:
 - **numberOfWheels** = 4
 - **engineCapacity** = 5
 - Apabila **carType** bernilai **Bus**, maka atribut **testCar** akan menjadi sebuah objek baru dengan kelas **Bus** yang memiliki parameter:
 - **numberOfWheels** = 6
 - **engineCapacity** = 10
 - **passengerCapacity** = 20
 - **isUsedForLongTrip** = **true**
 - Apabila **carType** bernilai **Perari**, maka atribut **testCar** akan menjadi sebuah objek baru dengan kelas **Perari** yang memiliki parameter:
 - **numberOfWheels** = 4
 - **engineCapacity** = 20
 - **averageSpeed** = 50
 - **isInsuranced** = **false**
 - *Hint: Pengecekan kesamaan antara dua string sebaiknya menggunakan method equals() yang dimiliki kedua Object tersebut. Contoh: "test".equals("test") == true*
5. **printCarList**, yang akan melakukan iterasi terhadap isi **carList**, kemudian, untuk setiap elemennya, dia akan secara berturut-turut melakukan hal berikut:
 1. Tergantung *instance* dari elemennya:
 - Untuk kelas **Car** saja dan bukan *instance* dari childnya, maka cetak "**Car**: " lalu cetak output fungsi **sound()** dan **honk()** dengan semua kata dipisah oleh spasi diakhiri *new line*
 - Untuk kelas **Bus**, cetak "**Bus**: ", lalu cetak output fungsi **sound()** dan **honk()** dengan semua kata dipisah oleh spasi dan diakhiri *new line*
 - Untuk kelas **Perari**, cetak "**Perari**: ", lalu cetak output fungsi **sound()** dan **honk()** dengan semua kata dipisah oleh spasi diakhiri *new line*
 - Contoh untuk kelas **Bus**, maka output yang benar adalah : **Bus: Cesss Notnot**
 - *Hint: Pengecekan instance sebuah kelas dilakukan dengan menggunakan operator instanceof (Contoh: "string" instanceof String == true)*
 2. Memanggil metode **printDescription** diakhiri *new line*

Lengkapi file [CarPrinter.java](#)

Submit file **CarPrinter.java**

Java 8

 [CarPrinter.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	12	Accepted	0.13 sec, 29.02 MB
2	12	Accepted	0.10 sec, 28.07 MB
3	12	Accepted	0.09 sec, 29.88 MB
4	12	Accepted	0.08 sec, 28.05 MB
5	12	Accepted	0.09 sec, 30.70 MB
6	12	Accepted	0.09 sec, 28.09 MB
7	12	Accepted	0.09 sec, 28.07 MB
8	16	Accepted	0.13 sec, 29.34 MB

Q
3
C
M
10
ou
of
10

1. ☐ 2. ☐ 3. ☐ 4. ☐ 5. ☐ 6. ☐ 7. ☐ 8. ☐ 9. ☐ 10. ☐

1. The following code snippet is a valid Java program. What is the output of the program?

```
public class Main {
    public static void main(String[] args) {
        int x = 10;
        int y = 20;
        int z = x + y;
        System.out.println(z);
    }
}
```

1

2. The following code snippet is a valid Java program. What is the output of the program?

```
public class Main {
    public static void main(String[] args) {
        int x = 10;
        int y = 20;
        int z = x * y;
        System.out.println(z);
    }
}
```

1

3. The following code snippet is a valid Java program. What is the output of the program?

```
public class Main {
    public static void main(String[] args) {
        int x = 10;
        int y = 20;
        int z = x / y;
        System.out.println(z);
    }
}
```

1

4. The following code snippet is a valid Java program. What is the output of the program?

```
public class Main {
    public static void main(String[] args) {
        int x = 10;
        int y = 20;
        int z = x % y;
        System.out.println(z);
    }
}
```

1

