



Home > My courses > PROG. S1 FAK. REGULER > REG - Genap 2023/2024 > [Reg] Dasar-Dasar Pemrograman 2 (A,B,C,D,E,F,G) Ge... > Week 10: Abstract Classes and Interfaces [LQ] > Kuis Mingguan 7: Abstract Classes and Interfaces

**Started on** Thursday, 4 April 2024, 1:40 PM

**State** Finished

**Completed on** Thursday, 4 April 2024, 2:04 PM

**Time taken** 23 mins 41 secs

**Grade** 13.00 out of 15.00 (87%)

### Question 1

Partially correct

Mark 2.00 out of 3.00

Untuk setiap kriteria di bawah ini, pilihlah karakteristik yang tepat untuk *abstract class* dan *interface*.

- Memiliki *non-abstract methods*

Abstract class  ✓

Interface  ✗

- Memiliki *constructor*

Abstract class  ✗

Interface  ✓

- Sebuah kelas bisa memiliki  ✓ *inheritance* dan  ✓ *extentions* (interface).

## Question 2

Correct

Mark 3.00 out of 3.00

Perhatikan potongan kode berikut ini:

```
1 interface TestInterface{
2     public static int value = 100;
3     public void display();
4 }
5
6 class TestClass implements TestInterface{
7     public static int value = 5000;
8     public void display() {
9         System.out.println("This displays something");
10    }
11 }
12
13 public class Main{
14     public static void main(String args[]) {
15         TestClass testObj = new TestClass();
16         TestInterface testInt = new TestClass();
17
18         System.out.println(TestInterface.value);
19         System.out.println(testObj.value);
20         System.out.println(testInt.value);
21     }
22 }
```

Apa keluaran dari program tersebut pada baris:

Line#18: 100 ✓

Line#19: 5000 ✓

Line#20: 100 ✓

## Question 3

Correct

Mark 1.00 out of 1.00

Perhatikan potongan kode di bawah ini:

```
1 abstract class AbstractClass {
2     abstract void abstractMethod() {
3         System.out.println("First Method");
4     }
5 }
6
```

Apa yang terjadi pada potongan kode tersebut?

Select one:

- ☒ a. Terjadi *compile error*, karena *abstractMethod()* memiliki implementasi ✓
- ☐ b. Terjadi *compile error*, karena *abstractMethod()* tidak dideklarasikan *public*
- ☐ c. Tidak akan terjadi *compile error*
- ☐ d. Terjadi *compile error*, karena *AbstractClass* tidak dideklarasikan *public*

Your answer is correct.

The correct answer is: Terjadi *compile error*, karena *abstractMethod()* memiliki implementasi

#### Question 4

Correct

Mark 1.00 out of 1.00

Diberikan potongan kode berikut:

```
1 public abstract class Orang {}
2
3 public abstract class Pelajar {}
4
5 public interface Berprestasi {}
6
7 public interface Berbakti {}
8
```

Pilihlah opsi potongan kode di bawah ini yang akan menghasilkan *error*!

Select one:

- ☐ a. `public class Mahasiswa extends Orang implements Berprestasi, Berbakti {}`
- ☒ b. `public class Mahasiswa extends Orang, Pelajar implements Berprestasi {}` ✓
- ☐ c. `public abstract class Mahasiswa extends Orang {}`
- ☐ d. `public interface Sukses extends Berprestasi, Berbakti {}`

Your answer is correct.

The correct answer is: `public class Mahasiswa extends Orang, Pelajar implements Berprestasi {}`

#### Question 5

Correct

Mark 1.00 out of 1.00

Diberikan potongan kode di bawah ini:

```
1 public abstract class Pelajar {}
2
3 public interface Berprestasi {}
4
5 public abstract class Mahasiswa extends Pelajar implements Berprestasi {}
6
```

Bagaimana kah cara instansiasi *object* yang tidak menghasilkan *error*?

Select one:

- ☐ a. `Berbakti m = new Mahasiswa();`
- ☐ b. `Pelajar m = new Pelajar();`
- ☐ c. `Mahasiswa m = new Pelajar();`
- ☒ d. `Pelajar m = new Mahasiswa();` ✓

Your answer is correct.

The correct answer is: `Pelajar m = new Mahasiswa();`

**Question 6**

Correct

Mark 1.00 out of 1.00

```
1 interface Kuis {  
2     final int k = 8;  
3     private int m;  
4     public void aMethod();  
5     String fun(String s);  
6 }
```

Baris ke berapa yang menghasilkan *error*?

Select one:

- ☒ a. 3 ✓
- ☐ b. 4
- ☐ c. 5
- ☐ d. 2

Your answer is correct.

The correct answer is: 3

**Question 7**

Correct

Mark 1.00 out of 1.00

Perhatikan potongan kode berikut ini:

```
public interface Berprestasi {  
    public <--JAWABAN1--> void belajar() {  
        System.out.println("Saya belajar setiap hari");  
    }  
  
    public <--JAWABAN2--> void bersosial() {  
        System.out.println("Menjaga koneksi");  
    }  
  
    public <--JAWABAN3--> void berorganisasi();  
}  
  
public class Mahasiswa implements Berprestasi {  
  
    @Override  
    public void bersosial() {  
        System.out.println("Sosialisasi agar tidak bosan");  
    }  
  
    @Override  
    public void berorganisasi() {  
        System.out.println("Berorganisasi menambah pengalaman");  
    }  
}
```

Apa isian yang tepat untuk JAWABAN1, JAWABAN2, JAWABAN3 secara berurutan?

Select one:

- ☐ a. static, static, abstract
- ☐ b. default, static, abstract
- ☐ c. static, abstract, abstract
- ☒ d. static, default, abstract ✓

Your answer is correct.

The correct answer is: static, default, abstract

**Question 8**

Incorrect

Mark 0.00 out of 1.00

Perhatikan bagan UML dari *abstract class Felis* (genus dari kucing) bawah ini.



Jika kita ingin membuat *child class FelisCatus* dari *Felis*, manakah implementasi yang benar?

Select one:

- ☐ a. 

```
public class FelisCatus extends Felis {  
    protected void bersuara() {  
        System.out.println("Miaow");  
    }  
  
    public void displayInfo() {  
        System.out.println("This is felis catus");  
    }  
}
```
- ☐ b. 

```
public class FelisCatus extends Felis {  
    void bersuara() {  
        System.out.println("Miaow");  
    }  
  
    protected void makan() {  
        System.out.println("Purr");  
    }  
}
```
- ☒ c. 

```
public class FelisCatus extends Felis {  
    void bersuara() {  
        System.out.println("Miaow");  
    }  
  
    void makan() {  
        System.out.println("Purr");  
    }  
  
    public void displayInfo() {  
        System.out.println("This is felis catus");  
    }  
}
```

 ✖
- ☐ d. 

```
public class FelisCatus extends Felis {  
    protected void makan() {  
        System.out.println("Purr");  
    }  
  
    public void displayInfo() {  
        System.out.println("This is felis catus");  
    }  
}
```

Your answer is incorrect.

The correct answer is:

```
public class FelisCatus extends Felis {  
    void bersuara() {  
        System.out.println("Miaow");  
    }  
  
    protected void makan() {  
        System.out.println("Purr");  
    }  
}
```

**Question 9**

Correct Mark 1.00 out of 1.00

Jika diberikan sebuah potongan kode dari interface Berprestasi di bawah ini.

```
public interface Berprestasi {  
    int jam_belajar = 8;  
  
    abstract void belajar();  
    abstract void berorganisasi();  
}
```

Manakah di bawah ini yang berhasil mengimplementasikan Berprestasi dengan benar?

Select one:

- ☐ a. 

```
public class Mahasiswa implements Berprestasi {  
    @Override  
    public static void belajar() {  
        System.out.println("Saya harus belajar setiap hari");  
    }  
}
```
- ☒ b. 

```
public class Mahasiswa implements Berprestasi {  
    @Override  
    public void belajar() {  
        System.out.println("Saya harus belajar setiap hari");  
    }  
  
    @Override  
    public void berorganisasi() {  
        System.out.println("Selalu aktif berorganisasi");  
    }  
}
```

 ✓
- ☐ c. 

```
public class Mahasiswa implements Berprestasi {  
    @Override  
    void berorganisasi() {  
        System.out.println("Selalu aktif berorganisasi");  
    }  
}
```
- ☐ d. 

```
public class Mahasiswa implements Berprestasi {  
    @Override  
    void belajar() {  
        System.out.println("Saya harus belajar setiap hari");  
    }  
  
    @Override  
    void berorganisasi() {  
        System.out.println("Selalu aktif berorganisasi");  
    }  
}
```

Your answer is correct.

```
public class Mahasiswa implements Berprestasi {  
    @Override  
    public void belajar() {  
        System.out.println("Saya harus belajar setiap hari");  
    }  
  
    @Override  
    public void berorganisasi() {  
        System.out.println("Selalu aktif berorganisasi");  
    }  
}
```

The correct answer is:



### Question 10

Correct

Mark 1.00 out of 1.00

```
public class Mahasiswa implements Cloneable {}
```

Potongan kode di atas akan menghasilkan *error*, karena *method* `clone()` belum di-override.

Select one:

- ☐ True
- ☒ False ✓

The correct answer is 'False'.

### Question 11

Correct

Mark 1.00 out of 1.00

```
public class Mahasiswa implements Comparable {  
    @Override  
    public boolean compareTo(Object o) {  
        return true;  
    }  
}
```

Potongan kode di atas menghasilkan *compile error*.

Select one:

- ☒ True ✓
- ☐ False

The correct answer is 'True'.