



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 16 - 17: UAS > Ujian Akhir Semester  
DDP2 Genap 23/24

**Started on** Thursday, 30 May 2024, 9:15 AM

**State** Finished

**Completed on** Thursday, 30 May 2024, 11:43 AM

**Time taken** 2 hours 27 mins

**Grade** 73.00 out of 104.00 (70%)

### Question 1

Incorrect

Mark 0.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 2, 3, 4)

Jika superclass dan subclass berada pada Package yang sama, maka modifier yang dapat digunakan agar subclass dapat mengakses properties dan method yang dimiliki oleh superclass adalah modifier?

Select one:

- ☐ a. Public, Private, Default
- ☐ b. Public, Protected, Default, Private
- ☒ c. Public, Protected ✖
- ☐ d. Public, Protected, Default

Your answer is incorrect.

The correct answer is: Public, Protected, Default

**Question 2**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 3, 4, 5)

```
1 public class Animal {  
2     String name;  
3     int phone;  
4     void call(){  
5         System.out.println("Katakan Peta");  
6     }  
7  
8     public static void main(String[] args) {  
9         Croco hai = new Croco();  
10        System.out.println(hai);  
11    }  
12 }  
13  
14 class Croco extends Animal{  
15     private void call(){  
16         System.out.println("No Hape Kamu Berapa?");  
17     }  
18 }
```

Perhatikan code diatas, baris yang menyebabkan error adalah?

Select one:

- ☒ a. 15 ✓
- ☐ b. 4
- ☐ c. 8
- ☐ d. 9

Your answer is correct.

The correct answer is: 15

**Question 3**

Incorrect Mark 0.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 3, 4, 5)

```
1 public class Animal {
2     String name;
3     int phone;
4     void call(){
5         System.out.println("Katakan Peta");
6     }
7
8     public static void main(String[] args) {
9         Croco hai = new Croco();
10        System.out.println(hai);
11        Garangan hei = new Garangan();
12        System.out.println(hei);
13    }
14 }
15
16 class Croco extends Animal{
17     private void call(){
18         System.out.println("No Hape Kamu Berapa?");
19     }
20 }
21
22 class Cat extends Animal(){
23     void call(){
24         System.out.println("Meooooong");
25     }
26 }
27
28
29
30 class Garangan{
31     void call(){
32         System.out.println("Kalo aku Telfon, ada yang marah nggak?");
33     }
34 }
```

Perhatikan code diatas, method yang tidak melakukan override terdapat pada baris?

Select one:

- ☒ a. 9 ✖
- ☐ b. 16
- ☐ c. 23
- ☐ d. 31

Your answer is incorrect.

The correct answer is: 31

#### Question 4

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 2, 3, 4)

Pernyataan yang **SALAH** tentang Abstract Class dan Abstract Method adalah?

Select one:

- ☐ a. Abstract Class tidak dapat diinstansiasi.
- ☐ b. Abstract Method harus diimplementasikan pada subclassnya yang konkrit.
- ☐ c. Abstract Class boleh memiliki abstract method dan concrete method.
- ☒ d. Abstract Method tidak harus diimplementasikan pada subclassnya yang konkrit. ✓

Your answer is correct.

The correct answer is: Abstract Method tidak harus diimplementasikan pada subclassnya yang konkrit.

**Question 5**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 2, 3, 4)

```
1 public abstract class Animal {
2     String name;
3     int phone;
4     void call(){
5         System.out.println("Katakan Peta");
6     }
7     abstract void walk(){
8         System.out.println("Dijemput sesuai Peta");
9     }
10
11     public static void main(String[] args) {
12         Croco hai = new Croco();
13         System.out.println(hai);
14     }
15 }
16
17 class Croco extends Animal{
18     private void call(){
19         System.out.println("No Hape Kamu Berapa?");
20     }
21
22     void walk(){
23         System.out.println("Aku jemput depan gang ya");
24     }
25 }
```

Perhatikan Code diatas, baris yang menyebabkan error adalah baris?

Select one:

- ☐ a. 22 dan 7
- ☐ b. 18 dan 22
- ☒ c. 18 dan 7 ✓
- ☐ d. 4 dan 7

Your answer is correct.

The correct answer is: 18 dan 7

**Question 6**

Incorrect

Mark 0.00 out of 2.00

(Soal ini menguji CMPK1: Sub-CPMK 4,5)

Program berikut ini akan mencetak tulisan **Selamat mengerjakan UAS DDP2 :)**

Isian yang tepat untuk \_\_(2)\_\_ pada program berikut ini adalah ...

```
interface InterfaceX{
    default void m1(){
        System.out.print("Selamat ");
    }

    String m2();
}

abstract class AbstractClassY{
    abstract String m3();
}

class ConcreteClass extends AbstractClassY implements InterfaceX{
    __(1)_{
        System.out.print("Selamat mengerjakan ");
    }

    public String m2(){
        return "UAS DDP2 " + m3();
    }

    String m3(){
        return ":";
    }
}

public class Test {
    public static void main(String[] args) {
        __(2) object= new ConcreteClass();
        object.m1();
        System.out.println(object.m2());
    }
}
```

Select one:

- ☐ a. InterfaceX
- ☐ b. AbstractClassY
- ☐ c. String
- ☒ d. Object ✖

Your answer is incorrect.

The correct answer is: InterfaceX

**Question 7**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CMPK1: Sub-CPMK 4,5)

Agar program berikut ini dapat dicompile dengan sukses, maka isian yang benar untuk \_\_(1)\_\_ adalah ...

```
interface InterfaceX{
    default void m1(){
        System.out.print("Selamat ");
    }

    String m2();
}

abstract class AbstractClassY{
    abstract String m3();
}

class ConcreteClass extends AbstractClassY implements InterfaceX{
    __(1)__ {
        System.out.print("Selamat mengerjakan ");
    }

    public String m2(){
        return "UAS DDP2 " + m3();
    }

    String m3(){
        return ":";
    }
}
```

Select one:

- ☐ a. default void m1()
- ☒ b. public void m1() ✓
- ☐ c. void m1()
- ☐ d. default public void m1()

Your answer is correct.

The correct answer is: public void m1()

**Question 8**

Incorrect

Mark 0.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 4, 5, 6)

Pernyataan berikut ini merupakan pernyataan yang benar mengenai objek dari kelas File dalam Java, **kecuali** ...

Select one:

- ☐ a. Objek kelas File dapat menghapus suatu file
- ☒ b. Objek kelas File dapat membuat suatu folder ❌
- ☐ c. Objek kelas file dapat melakukan write pada file
- ☐ d. Objek kelas file dapat mengupdate nama dari file

Your answer is incorrect.

The correct answer is: Objek kelas file dapat melakukan write pada file

**Question 9**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 4, 5, 6)

Apa output dari program di bawah ini ?

```
class A {}
class B extends A {}
class C extends B {}

public class Test {
    public static void main(String[] args) {
        try{
            A a = new B();
            C c = (C) a;
            int d = 10/0;

        }catch(ClassCastException e){
            System.out.println("error 1");
        }catch(RuntimeException e){
            System.out.println("error 2");
        }catch(ArithmeticException e){
            System.out.println("error 3");
        }
    }
}
```

Select one:

- ☒ a. Compile error ✔️
- ☐ b. Mencetak "error 1"
- ☐ c. Mencetak "error 3"
- ☐ d. Mencetak "error 2"

Your answer is correct.

The correct answer is: Compile error



**Question 10**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 4, 5, 6)

Apa output dari program berikut ini ?

```
class A {}
class B extends A {}
class C extends B {}

public class Test {
    public static void main(String[] args) {
        try{
            A a = new C();
            B b = (B) a;
            int d = 10/0;

        }catch(ClassCastException e){
            System.out.println("error 1");
        }catch(RuntimeException e){
            System.out.println("error 2");
        }
    }
}
```

Select one:

- ☐ a. Compile error
- ☐ b. Mencetak "error 1"
- ☒ c. Mencetak "error 2" ✓
- ☐ d. Tidak terjadi exception

Your answer is correct.

The correct answer is: Mencetak "error 2"

**Question 11**

Incorrect

Mark 0.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 4, 5, 6)

Perhatikan potongan code berikut ini. Urutan alur eksekusi program adalah \_\_\_\_

```
try {  
    // 1  
    throw new TypeNotPresentException(new String(), new Throwable());  
} catch(TypeNotPresentException e) {  
    // 2  
    throw new NullPointerException();  
} catch(NullPointerException e) {  
    // 3  
} catch(RuntimeException e) {  
    // 4  
} catch(Exception e) {  
    // 5  
} finally {  
    // 6  
}
```

Select one:

- ☐ a. 1-2-6
- ☐ b. 1-2
- ☒ c. 1-2-3-6 ✖
- ☐ d. 1-2-3

Your answer is incorrect.

The correct answer is: 1-2-6

**Question 12**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 4, 5, 6)

Bagaimana cara mendeklarasikan sebuah ArrayList yang hanya bisa menyimpan objek String?

Select one:

☐ a.

```
ArrayList<String> list = new ArrayList();
```

☐ b.

```
ArrayList<Object> list = new ArrayList<String>();
```

☒ c.

```
ArrayList<String> list = new ArrayList<String>();
```



☐ d.

```
ArrayList list = new ArrayList<String>();
```

Your answer is correct.

The correct answer is:

```
ArrayList<String> list = new ArrayList<String>();
```

**Question 13**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 4, 5, 6)

Mengapa kode dibawah ini tidak dapat dikompilasi:

```
public static <T> void nm(T t) {  
    System.out.println(t.getName());  
}
```

Select one:

- ☐ a. T tidak dapat digunakan sebagai parameter generik
- ☐ b. method nm harus mereturn suatu nilai
- ☐ c. method nm harus statis
- ☒ d. Tidak ada method getName() yang terdefinisi untuk tipe T ✓

Your answer is correct.

The correct answer is: Tidak ada method getName() yang terdefinisi untuk tipe T

**Question 14**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 4, 5, 6)

Bagaimana cara mendeklarasikan suatu method dengan nama nm yang dapat menerima argumen dengan tipe generik dan dapat mengembalikan nilai dengan tipe yang sama?

Select one:

☐ a.

`<T> nm(T arg)`

☒ b.

`<T> T nm(T arg)`



☐ c.

`T <T> nm(T arg)`

☐ d.

`T nm(T arg)`

Your answer is correct.

The correct answer is:

`<T> T nm(T arg)`

**Question 15**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 4, 5, 6)

Apa yang terjadi ketika kode dibawah ini dikompilasi?

```
class Mammal {}  
class Cat extends Mammal { }  
List<Mammal> list = new ArrayList<Cat>();
```

Select one:

- ☐ a. Error, karena tipe ArrayList bukan merupakan List
- ☐ b. Tidak error, karena tipe ArrayList meng-implement List
- ☐ c. Tidak error, karena tipe Cat meng-ekstend Mammal
- ☒ d. Error, karena tipe pada List mesti sama dengan pada ArrayList ✓

Your answer is correct.

The correct answer is: Error, karena tipe pada List mesti sama dengan pada ArrayList

**Question 16**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 7)

Bagian manakah pada kode berikut yang menyebabkan error pada saat kompilasi?

```
HBox hbox = new HBox();  
Label label = new Label("Name:");  
hbox.add(label);
```

Select one:

- ☐ a. HBox tidak mendukung Label
- ☒ b. Baris ketiga salah, kode yang benar adalah `hbox.getChildren().add(label);` ✓
- ☐ c. HBox harus diatur dulu ukurannya sebelum ditambahkan
- ☐ d. Label harus diatur dulu ukurannya sebelum ditambahkan

Your answer is correct.

The correct answer is: Baris ketiga salah, kode yang benar adalah `hbox.getChildren().add(label);`

**Question 17**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 7)

Apakah luaran (output) dari kode berikut saat dikompilasi dan dieksekusi?

```
TextField textField = new TextField();  
textField.setOnAction(e -> {  
    String text = textField.getText();  
    System.out.println(text);  
});
```

Select one:

- ☒ a. Teks yang diinput dengan keyboard akan ditampilkan pada standar output saat tombol Enter ditekan ✓
- ☐ b. Error pada saat kompilasi karena event handler tidak diatur dengan benar
- ☐ c. Teks yang diinput pada keyboard akan ditampilkan pada standar output setiap kali ada perubahan
- ☐ d. Tidak ada output karena event handler tidak diatur dengan benar

Your answer is correct.

The correct answer is: Teks yang diinput dengan keyboard akan ditampilkan pada standar output saat tombol Enter ditekan

**Question 18**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 7)

Deskripsi yang paling tepat untuk menggambarkan Scene dalam JavaFX adalah:

Select one:

- ☐ a. Kontainer untuk mengatur tataletak (layout).
- ☐ b. Struktur data untuk menyimpan event
- ☒ c. Kontainer utama pada window dimana elemen GUI dapat ditempatkan ✓
- ☐ d. Jendela utama pada JavaFX

Your answer is correct.

The correct answer is: Kontainer utama pada window dimana elemen GUI dapat ditempatkan

**Question 19**

Incorrect

Mark 0.00 out of 2.00

Mengapa menggunakan bahasa FXML untuk user interface lebih baik dibanding menggunakan Java secara langsung?

Select one:

- ☐ a. FXML memungkinkan aplikasi berjalan lebih cepat
- ☒ b. FXML memungkinkan akses langsung ke API database ✗
- ☐ c. FXML memungkinkan pemisahan yang jelas antara logika bisnis dan tampilan
- ☐ d. FXML memungkinkan penggunaan lebih sedikit memori

Your answer is incorrect.

The correct answer is: FXML memungkinkan pemisahan yang jelas antara logika bisnis dan tampilan

**Question 20**

Correct

Mark 2.00 out of 2.00

(Soal ini menguji CPMK1: Sub-CPMK 7)

Komponen JavaFX yang dapat digunakan untuk menangani tata letak (layout) adalah:

Select one:

- ☐ a. Button
- ☐ b. EventHandler
- ☒ c. VBox ✓
- ☐ d. TextField

Your answer is correct.

The correct answer is: VBox

**Question 21**

Correct

Mark 2.00 out of 2.00

**(Soal ini menguji CPMK1: Sub-CPMK 4,5)**

Apa yang akan terjadi ketika `Thread.yield()` dipanggil ketika sebuah thread dieksekusi?

Select one:

- ☐ a. Thread akan dihentikan secara permanen
- ☐ b. Thread akan mengubah prioritasnya menjadi prioritas terendah
- ☐ c. Thread akan berhenti dan keluar dari method `run()`
- ☒ d. Thread akan menghentikan eksekusi sehingga thread lain dapat dieksekusi ✓

Your answer is correct.

The correct answer is: Thread akan menghentikan eksekusi sehingga thread lain dapat dieksekusi

**Question 22**

Incorrect

Mark 0.00 out of 2.00

**(Soal ini menguji CPMK1: Sub-CPMK 4,5)**

Bagaimana cara memulai thread pada kode Java berikut?

```
Thread t = new Thread(new MyRunnable());
```

Select one:

- ☒ a.

`t.run();`



- ☐ b.

`t.execute();`

- ☐ c.

`t.start();`

- ☐ d.

`t.begin();`

Your answer is incorrect.

The correct answer is:

`t.start();`

**Question 23**

Correct

Mark 12.00 out of 12.00

(Soal ini menguji CMPK1: Sub-CPMK 3,4,5)

Perhatikan potongan code berikut ini:

```
1  import java.util.*;
2
3  public class CreateDrinkMenu{
4      public static void main(String[] args){
5          ArrayList<__(A)> drinkMenu= new ArrayList<>();
6          drinkMenu.add(new Drink(200,"Locacola"));
7          drinkMenu.add(new Coffee(150, "Cappuccino", 2));
8          drinkMenu.add(new Coffee(100, "Macchiato", 1));
9          drinkMenu.add(new Tea(200, "Camomile"));
10         drinkMenu.add(new Tea(100, "Earl Grey"));
11
12         for(__(A) myDrink : drinkMenu){
13             if(myDrink instanceof __(B))
14                 ((__(B))myDrink).brew();
15             myDrink.drinkIt();
16         }
17     }
18 }
19
20 class Drink{
21     __(C) double volume;
22     __(C) String name;
23     public Drink(){ }
24     public Drink(double volume, String name){
25         this.volume = volume;
26         this.name = name;
27     }
28     public void drinkIt(){
29         System.out.printf("Glek glek glek\n");
30     }
31 }
32
33 class Coffee extends Drink{
34     protected int numEspressoShots;
35     public Coffee(double volume, String name, int numEspressoShots){
36         this.name = name;
37         this.volume = volume;
38         this.numEspressoShots = numEspressoShots;
39     }
40     public void brew(){
41         System.out.printf("Brewing %d shots of espresso\n", this.numEspressoShots);
42     }
43     public void drinkIt(){
44         System.out.printf("Sipping %s\n", this.name);
45     }
46 }
47
48 class Tea extends Drink{
49     public Tea(double volume, String name){
50         super(volume, name);
51     }
52 }
```

Output yang diharapkan adalah:



```
Glek glek glek
Brewing 2 shots of espresso
Sipping Cappuccino
Brewing 1 shots of espresso
Sipping Macchiato
Glek glek glek
Glek glek glek
```

Isilah jawaban isian tanpa spasi, case sensitive (perhatian huruf kapital).

1. Lengkapi potongan code di atas pada bagian (A), (B), dan (C):

A.  ✓

B.  ✓

C.  ✓

2. Pada potongan code tersebut, terdapat  ✓ kali pemanggilan super, yang terdiri dari  ✓ kali pemanggilan eksplisit dan  ✓ kali pemanggilan implisit.

3. Pada potongan code tersebut, terdapat  ✓ kali overloading, salah satunya adalah pada method  ✓. (Tulis nama atau signature method. Jika lebih dari 1, tulis salah satu saja. Jika tidak ada, tuliskan none).

**Question 24**

Partially correct

Mark 4.00 out of 6.00

(Soal ini menguji CMPK1: Sub-CPMK 3,5)

Berdasarkan potongan kode di bawah ini, lengkapilah jawaban pada bagian yang telah disediakan!

```
1 abstract class Food {
2     String name;
3
4     public Food(String name) {
5         this.name = name;
6     }
7
8     public <JAWABAN> void spicyWarning() {
9         System.out.println("Warning! This food is spicy!");
10    }
11
12    public abstract void ingredients();
13    public abstract void displayInfo();
14 }
15
16 abstract class Dish extends Food {
17
18     public Dish(String name) {
19         super(name);
20         System.out.println("O'dish has been delightful");
21     }
22
23     public void displayInfo() {
24         System.out.printf("A meal named %s %n", this.name);
25     }
26
27     public abstract void howToEat();
28 }
29
```

1. Jika kita ingin membuat sebuah *concrete class* **Pasta** yang meng-*extend* class **Dish** harus meng-*override* sebanyak minimal



*methods*. (Isilah dengan angka jumlah method, atau isi none jika tidak ada).

2. Jika diketahui terdapat sebuah *modifier* yang mengisi kolom <JAWABAN> (dipastikan bukan *no modifier*), maka *modifier* yang mungkin adalah



(tanpa spasi, *case sensitive*).

```
1 class Main {
2     public static void main(String[] args) {
3         Dish[] dishes = new Dish[3];
4         dishes[1] = new Dish("Fettuccine Alfredo");
5         dishes[1].displayInfo();
6     }
7 }
8
```

3. Jika program di atas dijalankan, baris yang akan menghasilkan *error* adalah baris nomor  . (Isilah dengan angka nomor baris, atau isi none jika kode tidak menghasilkan *error* sama sekali).

1. Minimal methods yang harus di-*override* adalah **ingredients()** dan **howToEat()**, adapun **displayInfo()** tidak perlu di-*override* dikarenakan sudah ada implementasinya di *class Dish*.

2. Bisa mengisi di antara dua, yaitu **final** dan **static**. Tanpa menambahkan *modifier* masih memungkinkan namun di soal sudah dipastikan bukan *no modifier*. Modifier default tidak berlaku dalam abstract class dan hanya berlaku dalam interface.

3. Baris nomor 4 menghasilkan *compile error* dikarenakan abstract class **Dish** tidak bisa diinstansiasi.

**Question 25**

Complete

Mark 3.00 out of 6.00

(Soal ini menguji CMPK1: Sub-CPMK 3,5)

Berdasarkan potongan kode di soal sebelumnya, jawablah pertanyaan di bawah ini!

```
1 abstract class Food {  
2     String name;  
3  
4     public Food(String name) {  
5         this.name = name;  
6     }  
7  
8     public abstract void ingredients();  
9     public abstract void displayInfo();  
10 }  
11  
12 abstract class Dish extends Food {  
13  
14     public Dish(String name) {  
15         super(name);  
16         System.out.println("O'dish has been delightful");  
17     }  
18  
19     public void displayInfo() {  
20         System.out.printf("A meal named %s %n", this.name);  
21     }  
22  
23     public abstract void howToEat();  
24 }  
25
```

Apa saja perubahan yang perlu diterapkan untuk mengubah abstract class **Food** menjadi sebuah interface?

Hint: cukup sebutkan perubahan pada potongan kode di atas sehingga dapat berjalan tanpa *error*.

1. (line 2) modifier nya seharusnya public saja
2. (line 4-6) constructor dihapus saja, soalnya interface tidak boleh ada constructor
3. (line 12) extends diubah menjadi implements, karena itu kata kunci ketika ingin implement sebuah interface, kecuali jika Dish merupakan sebuah interface juga, baru Dish bisa memakai kata kunci extends

**Line 1:** Mengubah abstract class menjadi interface

**Line 2:** Menginisiasi/menghapus/memindahkan variabel String name

**Line 4-6:** Menghapus constructor

**Line 12:** Mengubah kata kunci extends menjadi implements

**Line 15:** Menghapus/mengganti kata kunci super(name) untuk constructor

**Catatan tambahan:**

- *Modifier* yang tertulis di method dan variabel tidak menjadi masalah, karena interface akan secara otomatis membacanya sebagai **public abstract** dan **public static final** (perhatikan bahwa ketika nilainya final, diperlukan instansiasi nilai variabel).
- Abstract class **Dish** tidak perlu mengimplementasikan semua abstract method dari **Food**, karena abstract class masih boleh memiliki abstract method.

Comment:

**Question 26**

Partially correct

Mark 11.00 out of 12.00

(Soal ini menguji CMPK1: Sub-CPMK 3,4,5)

Perhatikan potongan kode di bawah ini!

```
1 abstract class Food {}
2
3 interface SugarLevel {
4     public static void warning() {
5         System.out.println("Don't consume too much sugar");
6     }
7
8     public abstract void sugarContent();
9
10    public <JAWABAN 1> void sugarOption() {
11        System.out.println("Sugar level options: low, medium, high");
12    }
13 }
14
15 interface Cutlery {
16     void requiredCutlery();
17 }
18
19 class Dessert <JAWABAN 2> Food <JAWABAN 3> Cutlery, SugarLevel {
20     <JAWABAN 4> void requiredCutlery() {
21         System.out.println("Requires a fork and spoon.");
22     }
23
24     public void <JAWABAN 5> {
25         System.out.println("Sugar content is medium.");
26     }
27
28     public void sugarOption() {
29         System.out.println("Sugar options: level 1, level 2, level 3, level 4");
30     }
31 }
32
```

Berikan isian yang tepat untuk potongan kode yang belum lengkap tersebut (*case sensitive*).

Tulis "none" (tanpa kutip) jika tidak ada potongan kode yang perlu ditambahkan.

JAWABAN 1:  ✓

JAWABAN 2:  ✓

JAWABAN 3:  ✓

JAWABAN 4:  ✓

JAWABAN 5:  ✓

```
1 public class Main {
2     public static void main(String[] args) {
3         SugarLevel beverage = new Dessert();
4         beverage.sugarOption();
5     }
6 }
7
```

Output dari program tersebut adalah  ✓ (tanpa kutip).

Tulis "error" (tanpa kutip) jika program tidak berhasil dijalankan karena error.

**Question 27**

Partially correct

Mark 10.00 out of 12.00

(Soal ini menguji CMPK1: Sub-CPMK 5,6)

Perhatikan dua JAVA CLASS berikut dan lengkapi isian (a) -- (f) pada ke dua JAVA CLASS tersebut dengan ***statement*** yang paling akurat berdasarkan potongan program yang diberikan sehingga program dapat di-*compile/execute* dengan benar !

```
1. import java.util.Scanner;
2. import java.io.*;
3.
4. public class MainReadFile {
5.     public static void main(String[] args) {
6.         System.out.print("Enter a File Name: ");
7.         String fileName = new Scanner(System.in).next();
8.
9.         ReadFile fileReader = new ReadFile(fileName);
10.
11.         try {
12.             fileReader.readFile();
13.         }
14.         catch (_____ (a) ex) {
15.             System.out.println("First level exception");
16.         }
17.         catch (_____ (b) ex) {
18.             System.out.println("Next level exception");
19.         }
20.         catch (_____ (c) ex) {
21.             System.out.println("Other level exception");
22.         }
23.         finally {
24.             _____ (d);
25.         }
26.     }
27. }
```

```

1. import java.util.Scanner;
2. import java.io.*;
3.
4. public class ReadFile {
5.     private File myFile;
6.     private Scanner input;
7.
8.     public ReadFile(String fileName) {
9.         this.myFile = new _____(e);
10.    }
11.
12.    public void readFile() throws IOException, FileNotFoundException {
13.        int count = 0;
14.        this.input = new Scanner(this.myFile);
15.        while (_____(f)) {
16.            String line = input.nextLine();
17.            System.out.println(line);
18.            count += line.length();
19.        }
20.
21.        System.out.println("\nThe file size is " + count + " characters");
22.    }
23.
24.    public void close() {
25.        if (this.input != null)
26.            this.input.close();
27.    }
28. }

```

Tuliskan jawaban Anda dibawah ini (perhatikan setiap syntax dan penulisan jawaban, kesalahan pada penulisan yang berakibat pada compile error menunjukan kesalahan jawaban) !

isian (a)  ✓

isian (b)  ✓

isian (c)  ✓

isian (d)  ✓

isian (e)  ✓

isian (f)  ✗

**Question 28**

Partially correct

Mark 3.00 out of 12.00

(Soal ini menguji CMPK1: Sub-CPMK 5,6)

Perhatikan JAVA CLASS dibawah ini

```

1. import java.io.*;
2. import java.util.*;
3.
4. class Stack<E> {
5.     private ArrayList<E> A;
6.     private int top = -1;
7.     private int size;
8.
9.     public Stack(int size) {
10.         this.size = size;
11.         this.A = new ArrayList<E>(size);
12.     }
13.
14.     public void push(E x) {
15.         if (top + 1 == size) {
16.             System.out.println("Stack Overflow");
17.         } else {
18.             top = top + 1;
19.             if (A.size() > top)
20.                 A.set(top, x);
21.             else
22.                 A.add(x);
23.         }
24.     }
25.
26.     public E pop() {
27.         E retVal = null;
28.         if (top == -1) {
29.             System.out.println("Stack Underflow");
30.         } else {
31.             retVal = A.get(top);
32.             top--;
33.         }
34.         return retVal;
35.     }
36.
37.     public boolean empty() {
38.         return top == -1;
39.     }
40. }
41.
42. public class MainDriver {
43.     public static void main(String[] args) {
44.         Stack<Integer> s1 = new Stack<>(2);
45.         s1.push(10);
46.         s1.push(20);
47.
48.         Stack<float> s2 = new Stack<>(2);
49.         s2.push(100.0f);
50.         s2.push(200.0f);
51.         System.out.println(isExist(s2, 200.0f));
52.     }
53.
54.     public static boolean isExist(Stack<Number> stack, double y) {
55.         boolean retVal = false;
56.         double item;
57.         do {
58.             item = (double) stack.pop();
59.         } while (item.equals item != y && !stack.empty());
60.
61.         if (item == y)
62.             retVal = true;
63.         return retVal;

```



```
64.    }
65. }
```

Pada program diatas terdapat 2 baris program yang akan mengakibatkan *comple error* dan satu baris program yang berakibat *runtime error*. Identifikasi baris *error*, *statement* terkait, dan perbaikan yang diperlukan untuk menghilangkan error tersebut.

Contoh : misal pada baris 10 terdapat statement penyebab error "int myInt = 95.0f", maka pada jawaban, Anda harus menuliskan sebagai berikut (tidak menuliskan keseluruhan statement pada satu baris):

Baris Ke-	Statement	Perbaikan
10	95.0f	95

Tuliskan jawaban Anda dibawah ini:

(Jawaban ini akan diperiksa kembali secara manual.)

#### Compile Errors

Baris Ke-	Statement	Perbaikan
22 ✗	A.add(x); ✗	dihapus saja atau bisa juga System.out.println("Stack Overflow"); ✗
27 ✗	E retVal = null; ✗	E retVal; ✗

#### Runtime Error

Baris Ke-	Statement	Perbaikan
59 ✗	!item.equals ✗	!item.equals(y) ✗

Comment:

Bonus +3