**Latihan Abstract Class & interface**

# KETENTUAN

* **Tulis hasil dan penjelasan atas jawaban dari pertanyaan berikut (1-20)**
* **Tulis dalam format doc**
* **format nim\_nama\_lat\_abstract\_class\_interface.docx**
* **dikirmkan ke link dropbox berikut<https://www.dropbox.com/request/OUy2Bvvd9cgLK3UM1BbM>**
* **dikumpulkan di akhir perkuliahan jam 17.30**

SOAL

1. **Below class ABC doesn’t have even a single abstract method, but it has been declared as abstract. Is it correct?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-abstract-classes/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | Jawaban no 1 :  Salah secara konsep, karena **abstract class paling tidak harusnya memiliki satu abstract methods**, walau ketika di compile dengan netbeans tetap bisa dijalankan  abstract class ABC  {  void firstMethod()  {  System.out.println("First Method");  }  void secondMethod()  {  System.out.println("Second Method");  }  } |

1. **Why the below class is showing compilation error?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-abstract-classes/)

Jawaban no 2:

Hal ini karena ada abstract methods yang berisikan perintah, sedangkan **abstract methods seharusnya kosong**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | abstract class AbstractClass  {  abstract void abstractMethod()  {  System.out.println("First Method");  }  } |

1. **Which class is instantiable? Class A or Class B?**

Jawaban no 3 :

Tentu saja yang B, karena b bukan abstract class sedangkan class A adalah abstract class, dan kita tahu kalau **abstract class tidak dapat di-instansiasi**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-abstract-classes/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | abstract class A  {  }  class B extends A  {  } |

1. **Below code snippet is showing compilation error? Can you suggest the corrections?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-abstract-classes/)

Jawaban no 4 :

Hal ini karena class B belum mengimplementasikan abstract methods dari A, sedangkan kita mengetahui kalau **class yg menerima warisan dari abstract clas haruslah mengimplementasikan semua methods** yang ada di dalam pewaris

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | abstract class A  {  abstract int add(int a, int b);  }  class B extends A  {  } |

1. **Is the following program written correctly? If yes, what value “result” variable will hold if you run the program?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-abstract-classes/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | abstract class Calculate  {  abstract int add(int a, int b);  }  public class MainClass  {  public static void main(String[] args)  {  int result = new Calculate()  {  @Override  int add(int a, int b)  {  return a+b;  }  }.add(11010, 022011);  } |

Jawaban no 5 :

Benar, program ditulis dengan benar. Lalu value resultnya bernilai 20235. Hal ini karena apabila kita menuliskan leading zero sebelum menulis sebuah nominal int. **Java akan menganggap itu adalah angka octal**, sehingga compiler akan menkonversi nilai ke dalam integer. Pada program ini, 22011 akan dianggap sebagai biangan octal dan diubah ke desimal

|  |  |
| --- | --- |
| 16  17  18  19 | }  Jawaban no 6 :  Salah di bagian private, seharunya public karena **abstract methods harus di implementasikan oleh class lain**, sedangkan jika bertipe private maka methods ini tidak dapat diakses oleh class lain |

1. **Can you identify the error in the below code?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-abstract-classes/)

|  |  |
| --- | --- |
| 1  2  3  4 | abstract class AbstractClass  {  private abstract int abstractMethod();  } |

1. **Is the below program written correctly? If yes, what will be the output?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-abstract-classes/2/)

Jawaban no 7 :

Benar, program akan mengeluarkan hasil

1

2

5

3

4

Hal ini karena **methods yang tidak memiliki nama punya kedudukan yang lebih tinggi dibandingkan konstruktor**. Sehingga program menjalankan methods yang tidak mempunyai nama terlebih dahulu, sebelum menjalankan konstruktor

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | abstract class XYZ  {  {  System.out.println(1);  }  public XYZ()  {  System.out.println(2);  abstractMethod();  }  abstract void abstractMethod();  }  class PQR extends XYZ  {  {  System.out.println(3);  }  public PQR()  {  System.out.println(4);  }  @Override  void abstractMethod()  {  System.out.println(5);  }  } |

|  |  |
| --- | --- |
| 30  31  32  33  34  35  36 | public class MainClass  {  public static void main(String[] args)  {  PQR pqr = new PQR();  }  }  Jawaban no 8 :  Bisa, program disamping eror, karena memiliki 2 kesalahan.   1. Sebuah class biasa, tidak boleh berisikan abstract methods. 2. Sebuah abstract methods haruslah kosong atau tidak memiliki body.   Bisa diperbaiki dengan **menghilangkan tulisan abstract** di mehods X |

1. **Can you identify the error in the below code?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | class X  {  public X()  {  System.out.println("Constructor One");  }  abstract X(int i)  {  System.out.println("Constructor Two");  }  } |

1. **Is the below code written correctly?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | class A  Jawaban no 9 :  Benar, **class dapat berada di dalam methods** suatu class  {  void methodOfA()  {  abstract class B  {  }  }  } |

1. **Is the below code written correctly?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-abstract-classes/3/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | class A  {  abstract class B  {  class C  {  abstract class D  {  abstract void method();  } |
|  |  |

Jawaban no 10 :

Bener, tidak ada hal yang membuat program ini eror. **Abstract class juga dapat berisikan class**-class lainnya

|  |  |
| --- | --- |
| 9  10  11  12  13 | }  }  } |

1. **Can you identify the error in the below code?**

Jawaban no 11 :

Bisa, program **eror karena menggunakan modifier private**, sedangkan interface akan di implements ke class lain, bagaimana bisa diimplement jika modifiernya private ?

Selain itu, **attribute pada interface harus di assign** oleh suatu nilai

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-interfaces/)

|  |  |
| --- | --- |
| 1  2  3  4 | interface A  {  private int i;  } |

1. **What will be the output of the following program?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-interfaces/)

Jawaban no 12 :

Outputnya “My Method”

Instansiasi

A a = new C() itu legal, karena **A sudah menjadi bagian dari C** akibat keyword implement dan extend. Sehingga ketika program dijalankan, akam memanggil muMethods() yang berisikan System.out.println(“My Method”)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | interface A  {  void myMethod();  }  class B  {  public void myMethod()  {  System.out.println("My Method");  }  }  class C extends B implements A  {  }  class MainClass  {  public static void main(String[] args)  {  A a = new C();  a.myMethod();  }  } |

1. **Why the below code is showing compile time error?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-interfaces/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | Jawaban no 13:  Hal ini karena **methodX tidak menggunakan public,** hal ini membuat methodsX tidak dapat diakses oleh interfacenya sendiri  interface X  {  void methodX();  }  class Y implements X  {  void methodX()  {  System.out.println("Method X");  }  } |

1. **Does below code compile successfully? If not, why?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-interfaces/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | Jawaban no 14:  Tentu saja **tidak bisa dicompile,** hal ini dikarenakan tidak adanya tipe data yang diberikan untuk variabel i.  interface A  {  int i = 111;  }  class B implements A  {  void methodB()  {  i = 222;  }  } |

1. **Is the following code written correctly?**

Jawaban no 15:

Program **tidak benar** karena interface **tidak bisa menerima warisan** dari **class**.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | class A  {  //Class A  }  interface B extends A  {  //Interface B extending Class A  } |

1. **Does below program compile successfully?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-interfaces/2/)

Jawaban no 16 :

Ya, walaupun interface tidak bisa menerima warisan dari class, tetapi **interface dapat menerima warisan dari interface** lainnya. Sehingga program disamping sudah benar.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | interface ABC  {  public void methodOne();  public void methodTwo();  }  interface PQR extends ABC  {  public void methodOne();  public void methodTwo();  } |

1. **Is the following program written correctly? If yes, what will be the output?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-interfaces/2/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | interface ABC  {  void methodOne();  }  interface PQR extends ABC  {  void methodTwo();  }  abstract class XYZ implements PQR  {  public void methodOne()  {  methodTwo();  }  }  class MNO extends XYZ  {  public void methodTwo()  {  methodOne();  }  }  public class MainClass  {  public static void main(String[] args)  {  ABC abc = new MNO();  abc.methodOne(); |

Jawaban no 17 :

Outputnya

Program **sudah benar**, dan tentu saja tidak ada outputnya karena tidak ada System.out.println(). Selain itu program ini juga tidak memanggil methodTwo() maupun methodONe()

|  |  |
| --- | --- |
| 29  30  31  32  33  34  35 | }  } |

1. **What will be the output of the following program?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-interfaces/2/)

Jawaban no 18 :

Outputnya

A

B

A

A

Baris pertama dikeluarkan ‘A’ akibat pemanggilan methods tanpa nama, baris kedua dikeluarkan ‘B’ akibat perintah return ++c. Tetapi baris ke 3 dan 4 tidak dipengaruhi oleh return ++c karena **yang di tambahkan hanyalah variabel lokal** dari methodX

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | interface X  {  char c = 'A';  char methodX();  }  class Y implements X  {  {  System.out.println(c);  }  public char methodX()  {  char c = this.c;  return ++c;  }  }  public class MainClass  {  public static void main(String[] args)  {  Y y = new Y(); System.out.println(y.methodX()); System.out.println(y.c);  System.out.println(X.c);  }  } |

1. **Can you identify the error in the below code?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-interfaces/2/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | Jawaban no 19 :  Eror berada pada program bagian interface C yang berada di dalam methodsA(). Kita tau bahwa **interface tidak bisa berada di dalam methods**  interface A  {  void methodA();  }  class B implements A  {  public void methodA()  {  interface C  {  int i = 123;  }  }  } |

1. **What will be the output of the following program?**

[?](http://javaconceptoftheday.com/java-practice-coding-questions-on-interfaces/3/)

Jawaban no 20 :

Outputnya

2

3

3

Untuk baris pertama dan kedua kita sudah tau kalau nilai ‘2’ dan ‘3’ keluar akibat perintah new A() dan new B(). Tetapi mengapa baris ketiga bernilai ‘3’ ? Hal ini karena ketika perintah new C()dijalankan, perintah method(B b) akan **dijalankan juga akibat extends** yang dilakuakan class C terhadap class B

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | class A { }  class B extends A { } class C extends B { }  interface ABC  {  void method(A a);  }  interface PQR  {  void method(B b);  }  class M implements ABC, PQR  {  public void method(A a)  {  System.out.println(2);  }  public void method(B b)  {  System.out.println(3);  }  } |

|  |  |
| --- | --- |
| 26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | public class MainClass  {  public static void main(String[] args)  {  M m = new M(); m.method(new A());  m.method(new B());  m.method(new C());  }  } |

**KETERANGAN**

* **Tulis hasil dan penjelasan atas jawaban dari pertanyaan berikut (1-20)**
* **Tulis dalam format doc**
* **format nim\_nama\_lat\_abstract\_class\_interface.docx**
* **dikirmkan ke link dropbox berikut <https://www.dropbox.com/request/OUy2Bvvd9cgLK3UM1BbM>**
* **dikumpulkan di akhir perkuliahan jam 17.30**