Praktikum

Polymorphism

LATIHAN SOAL

1. Which statement(s) are true? (Choose all that apply.)

   A. Has-a relationships always rely on inheritance.
   B. Has-a relationships always rely on instance variables.
   C. Has-a relationships always require at least two class types.
   D. Has-a relationships always rely on polymorphism.
   E. Has-a relationships are always tightly coupled.

2. Given:

```
class Clidders {
  public final void flipper() { System.out.println("Clidder"); }
}
public class Clidlets extends Clidders {
  public void flipper() {
    System.out.println("Flip a Clidlet");
    super.flipper();
  }
  public static void main(String [] args) {
    new Clidlets().flipper();
  }
}
```

What is the result?

   A. Flip a Clidlet
   B. Flip a Clidder
   C. Flip a Clidder

      Flip a Clidlet

   D. Flip a Clidlet

      Flip a Clidder

   E. Compilation fails.

3.
```
class Top {
  public Top(String s) { System.out.print("B"); }
}
public class Bottom2 extends Top {
```

```
  public Bottom2(String s) { System.out.print("D"); }
  public static void main(String [] args) {
    new Bottom2("C");
    System.out.println(" ");
  }
}
```

What is the result?

A. BD
B. DB
C. BDC
D. DBC
E. Compilation fails.

4. Given:

```
class Clidder {
  private final void flipper() { System.out.println ("Clidder"); }
}

public class Clidlet extends Clidder {
  public final void flipper() { System.out.println("Clidlet");  }
  public static void main(String [] args) {
    new Clidlet().flipper();
  }
}
```

What is the result?

A. Clidlet
B. Clidder
C. Clidder

   Clidlet

D. Clidlet

   Clidder

E. Compilation fails.

5. Given:

```
1. class Plant {
2.  String getName() { return "plant"; }
3.  Plant getType() { return this; }
4. }
5. class Flower extends Plant {
6.  // insert code here
7. }
8. class Tulip extends Flower {}
```

Which statement(s), inserted at line 6, will compile? (Choose all that apply.)

A. Flower getType() { return this; }
B. String getType() { return "this"; }
C. Plant getType() { return this; }
D. Tulip getType() { return new Tulip() ;}

6. Given:

```
1. class Zing {
2.   protected Hmpf h;
3. }
4. class Woop extends Zing { }
5. class Hmpf { }
```

Which is true? (Choose all that apply.)

A. Woop is-a Hmpf and has-a zing.
B. zing is-a Woop and has-a Hmpf.
C. Hmpf has-a Woop and Woop is-a Zing.
D. Woop has-a Hmpf and Woop is-a zing.
E. Zing has-a Hmpf and Zing is-a Woop.

7.

```
class Programmer {
2.   Programmer debug() { return this; }
3. }
4. class SCJP extends Programmer {
5.   // insert code here
6. }
```

Which, inserted at line 5, will compile? (Choose all that apply.)

A. Programmer debug() { return this; }
B. SCJP debug() { return this; }
C. Object debug() { return this; }
D. int debug() { return 1; }
E. int debug(int x) { return 1; }
F. Object debug (int x) { return this; }

8.

Given:

```
class Uber {
  static int y = 2;
  Uber(int x) { this(); y = y * 2; }
  Uber() { y++; }
}
class Minor extends Uber {
  Minor() { super(y); y = y + 3; }
```

```
  public static void main(String [] args) {
    new Minor();
    System.out.println(y);
} }
```

What is the result?

A. 6
B. 7
C. 8
D. 9
E. Compilation fails.
F. An exception is thrown.

9.

```
class Dog { }
 2. class Beagle extends Dog { }
 3.
 4. class Kennel {
 5.   public static void main(String [] arfs) {
 6.     Beagle bl = new Beagle();
 7.     Dog dogl = new Dog();
 8.     Dog dog2 = bl;
 9.     // insert code here
10. } }
```

Which, inserted at line 9, will compile? (Choose all that apply.)

A. Beagle b2 = (Beagle) dog1;
B. Beagle b3 = (Beagle) dog2;
C. Beagle b4 = dog2;
D. None of the above statements will compile.

10.

Given the following,

```
 1. class X { void dol() { } }
 2. class Y extends X { void do2() { } }
 3.
 4. class Chrome {
 5.   public static void main(String [] args) {
 6.     X x1 = new X();
 7.     X x2 = new Y();
 8.     Y y1 = new Y();
 9.     // insert code here
10. } }
```

Which, inserted at line 9, will compile? (Choose all that apply.)

A. x2.do2( );
B. (Y) x2. do2( );
C. ((Y)x2).do2();

```
   D. None of the above statements will compile.
```

11.

Given:

```
1. class Voop {
2.   public static void main(String [] args) {
3.     doStuff(1);
4.     doStuff(1, 2);
5.   }
6.   // insert code here
7. }
```

Which, inserted independently at line 6, will compile? (Choose all that apply.)

```
   A. static void doStuff(int... doArgs) { }
   B. static void doStuff (int [] doArgs) { }
   C. static void doStuff(int doArgs...) { }
   D. static void doStuff(int... doArgs, int y) { }
   E. static void doStuff(int x, int... doArgs) { }
```

12.

Given:

```
class Alien {
  String invade(short ships) { return "a few"; }
  String invade(short... ships) { return "many"; }
}
class Defender {
  public static void main(String [] args) {
    System.out.println(new Alien().invade(7));
  }
}
```

What is the result?

```
   A. many
   B. a few
   C. Compilation fails.
   D. The output is not predictable.
   E. An exception is thrown at runtime.
```

13.

Given:

```
class Eggs {
  int doX(Long x, Long y) { return 1; }
  int doX(long... x) { return 2; }
```

```
  int doX(Integer x, Integer y) { return 3; }
  int doX(Number n, Number m) { return 4; }
  public static void main(String[] args) {
    new Eggs().go();
  }
  void go () {
    short s = 7;
    System.out.print(doX(s,s) + " ");
    System.out.println(doX(7,7));
} }
```

What is the result?

A. 1 1
B. 2 1
C. 3 1
D. 4 1
E. 2 3
F. 3 3
G. 4 3

14.

Given:

```
class Mixer {
  Mixer() { }
  Mixer(Mixer m) { ml = m;}
  Mixer m1;
  public static void main(String[] args) {
    Mixer m2 = new Mixer();
    Mixer m3 = new Mixer(m2);  m3.go();
    Mixer m4 = m3.m1;          m4.go();
    Mixer m5 = m2.m1;          m5.go();
  }
  void go() { System.out.print("hi "); }
}
```

What is the result?

A. hi
B. hi hi
C. hi hi hi
D. Compilation fails
E. hi, followed by an exception
F. hi hi, followed by an exception

15.

Given:

```
class Fizz {
  int x = 5;
```

```
  public static void main(String[] args) {
    final Fizz f1 = new Fizz();
    Fizz f2 = new Fizz();
    Fizz f3 = FizzSwitch(f1,f2);
    System.out.println((f1 == f3) + " " + (f1.x == f3.x));
  }
  static Fizz FizzSwitch(Fizz x, Fizz y) {
    final Fizz z = x;
    z.x = 6;
    return z;
} }
```

What is the result?

A. true true
B. false true
C. true false
D. false false
E. Compilation fails.
F. An exception is thrown at runtime.

16.

Given:

```
class Knowing {
  static final long tooth = 343L;
  static long doIt(long tooth) {
    System.out.print(++tooth + " ");

    return ++tooth;
  }
  public static void main(String[] args) {
    System.out.print(tooth + " ");
    final long tooth = 340L;
    new Knowing().doIt(tooth);
    System.out.println(tooth);
  }
}
```

What is the result?

A. 343 340 340
B. 343 340 342
C. 343 341 342
D. 343 341 340
E. 343 341 343
F. Compilation fails.
G. An exception is thrown at runtime.

17.

Given:

```
class Bird {
  { System.out.print("bl "); }
  public Bird() { System.out.print("b2 "); }
}
class Raptor extends Bird {
  static { System.out.print("r1 "); }
  public Raptor() { System.out.print("r2 "); }
  { System.out.print("r3 "); }
  static { System.out.print("r4 "); }
}
class Hawk extends Raptor {
  public static void main(String[] args) {
    System.out.print("pre ");
    new Hawk();
    System.out.println("hawk ");
  }
}
```

What is the result?

A. pre b1 b2 r3 r2 hawk
B. pre b2 b1 r2 r3 hawk
C. pre b2 b1 r2 r3 hawk r1 r4
D. r1 r4 pre b1 b2 r3 r2 hawk
E. r1 r4 pre b2 b1 r2 r3 hawk
F. pre r1 r4 b1 b2 r3 r2 hawk
G. pre r1 r4 b2 b1 r2 r3 hawk
H. The order of output cannot be predicted.
I. Compilation fails.

18.

```
1. public interface A {
2.   public void doSomething(String thing);
3. }

1. public class AImpl implements A {
2.   public void doSomething(String msg) { }
3. }

1. public class B {
2.   public A doit() {
3.     // more code here
4.   }
5.
6.   public String execute() {
7.     // more code here
8.   }
9. }

1. public class C extends B {
2.   public AImpl doit() {
3.     // more code here
4.   }
5.
6.   public Object execute() {
7.     // more code here
8.   }
9. }
```

- A. Compilation will succeed for all classes and interfaces.
- B. Compilation of class C will fail because of an error in line 2.
- C. Compilation of class C will fail because of an error in line 6.
- D. Compilation of class AImpl will fail because of an error in line 2.

19.
Given:

```
1. public class Plant {
2.   private String name;
3.   public Plant(String name) { this.name = name; }
4.   public String getName() { return name; }
5. }
1. public class Tree extends Plant {
2.   public void growFruit() { }
3.   public void dropLeaves() { }
4. }
```

Which statement is true?

- A. The code will compile without changes.
- B. The code will compile if public Tree() { Plant(); } is added to the Tree class.
- C. The code will compile if public Plant() { Tree(); } is added to the Plant class.
- D. The code will compile if public Plant() { this("fern"); } is added to the Plant class.
- E. The code will compile if public Plant() { Plant("fern"); } is added to the Plant class.

20.

Given:

```
10. interface Foo {}
11. class Alpha implements Foo {}
12. class Beta extends Alpha {}
13. class Delta extends Beta {
14.   public static void main( String[] args ) {
15.     Beta x = new Beta();
16.     // insert code here
17.   }
18. }
```

Which code, inserted at line 16, will cause a java.lang.ClassCastException?

- A. Alpha a = x;
- B. Foo f = (Delta)x;
- C. Foo f = (Alpha)x;
- D. Beta b = (Beta)(Alpha)x;

21.

Given:

```
11. class A {
12.   public void process() { System.out.print("A,"); }
13. class B extends A {
14.   public void process() throws IOException {
15.     super.process();
16.     System.out.print("B,");
17.     throw new IOException();
18.   }
19. public static void main(String[] args) {
20.   try { new B().process(); }
21.   catch (IOException e) { System.out.println("Exception"); }}
```

What is the result?

- **A.** Exception
- **B.** A,B,Exception
- **C.** Compilation fails because of an error in line 20.
- **D.** Compilation fails because of an error in line 14.
- **E.** A NullPointerException is thrown at runtime.

22.

Given:

```
11. public class Yikes {
12.
13.  public static void go(Long n) {System.out.println("Long ");}
14.  public static void go(Short n) {System.out.println("Short ");}
15.  public static void go(int n) {System.out.println("int ");}
16.  public static void main(String [] args) {
17.    short y = 6;
18.    long z = 7;
19.    go(y);
20.    go(z);
21.  }
22. }
```

What is the result?

- **A.** int Long
- **B.** Short Long
- **C.** Compilation fails.
- **D.** An exception is thrown at runtime.

23.

```
10: public class Hello {
11:   String title;
12:   int value;
13:   public Hello() {
14:     title += " World";
15:   }
16:   public Hello(int value) {
17:     this.value = value;
18:     title = "Hello";
19:     Hello();
20:   }
21: }
```

and:

```
30: Hello c = new Hello(5);
31: System.out.println(c.title);
```

What is the result?

- A. Hello
- B. Hello World
- C. Compilation fails.
- D. Hello World 5
- E. The code runs with no output.
- F. An exception is thrown at runtime.

24.

Given:

```
1. class Super {
2.   private int a;
3.   protected Super(int a) { this.a = a; }
4. }
 ...
11. class Sub extends Super {
12.   public Sub(int a) { super(a); }
13.   public Sub() { this.a = 5; }
14. }
```

Which two, independently, will allow Sub to compile? (Choose two.)

- A. Change line 2 to:
  public int a;
- B. Change line 2 to:
  protected int a;
- C. Change line 13 to:
  public Sub() { this(5); }
- D. Change line 13 to:
  public Sub() { super(5); }
- E. Change line 13 to:
  public Sub() { super(a); }

25.

```
1. class ClassA {
2.   public int numberOfInstances;
3.   protected ClassA(int numberOfInstances) {
4.     this.numberOfInstances = numberOfInstances;
5.   }
6. }
7. public class ExtendedA extends ClassA {
8.   private ExtendedA(int numberOfInstances) {
9.     super(numberOfInstances);
10.   }
11.   public static void main(String[] args) {
12.     ExtendedA ext = new ExtendedA(420);
13.     System.out.print(ext.numberOfInstances);
14.   }
15. }
```

Which statement is true?

- A. 420 is the output.
- B. An exception is thrown at runtime.
- C. All constructors must be declared public.
- D. Constructors CANNOT use the private modifier.
- E. Constructors CANNOT use the protected modifier.

26.

Given:

```
1. interface A { public void aMethod(); }
2. interface B { public void bMethod(); }
3. interface C extends A,B { public void cMethod(); }
4. class D implements B {
5.   public void bMethod(){}
6. }
7. class E extends D implements C {
8.   public void aMethod(){}
9.   public void bMethod(){}
10.   public void cMethod(){}
11. }
```

What is the result?

- A. Compilation fails because of an error in line 3.
- B. Compilation fails because of an error in line 7.
- C. Compilation fails because of an error in line 9.
- D. If you define D e = new E(), then e.bMethod() invokes the version of bMethod() defined in Line 5.
- E. If you define D e = (D)(new E()), then e.bMethod() invokes the version of bMethod() defined in Line 5.
- F. If you define D e = (D)(new E()), then e.bMethod() invokes the version of bMethod() defined in Line 9.

27.

Given:

```
1. public class Base {
2.   public static final String FOO = "foo";
3.   public static void main(String[] args) {
4.     Base b = new Base();
5.     Sub s = new Sub();
6.     System.out.print(Base.FOO);
7.     System.out.print(Sub.FOO);
8.     System.out.print(b.FOO);
9.     System.out.print(s.FOO);
10.     System.out.print(((Base)s).FOO);
11. } }
12. class Sub extends Base {public static final String FOO="bar";}
```

What is the result?

- A. foofoofoofoofoo
- B. foobarfoobarbar
- C. foobarfoofoofoo
- D. foobarfoobarfoo
- E. barbarbarbarbar
- F. foofoofoobarbar
- G. foofoofoobarfoo

28.

Given classes defined in two different files:

```
1. package util;
2. public class BitUtils {
3.   public static void process(byte[]) { /* more code here */ }
4. }
```

```
1. package app;
2. public class SomeApp {
3.   public static void main(String[] args) {
4.     byte[] bytes = new byte[256];
5.     // insert code here
6.   }
7. }
```

What is required at line 5 in class SomeApp to use the process method of BitUtils?

- A. process(bytes);
- B. BitUtils.process(bytes);
- C. util.BitUtils.process(bytes);
- D. SomeApp cannot use methods in BitUtils.
- E. import util.BitUtils.*; process(bytes);

29.

Given:

```
13. public class Pass {
14.   public static void main(String [] args) {
15.     int x = 5;
16.     Pass p = new Pass();
17.     p.doStuff(x);
18.     System.out.print(" main x = " + x);
19.   }
20.
21.   void doStuff(int x) {
22.     System.out.print(" doStuff x = " + x++);
23.   }
24. }
```

What is the result?

- **A.** Compilation fails.
- **B.** An exception is thrown at runtime.
- **C.** doStuff x = 6 main x = 6
- **D.** doStuff x = 5 main x = 5
- **E.** doStuff x = 5 main x = 6
- **F.** doStuff x = 6 main x = 5

30.

Given:

```
11. static class A {
12.   void process() throws Exception { throw new Exception(); }
13. }
14. static class B extends A {
15.   void process() { System.out.println("B "); }
16. }
17. public static void main(String[] args) {
18.   A a = new B();
19.   a.process();
20. }
```

What is the result?

- **A.** B
- **B.** The code runs with no output.
- **C.** An exception is thrown at runtime.
- **D.** Compilation fails because of an error in line 15.
- **E.** Compilation fails because of an error in line 18.
- **F.** Compilation fails because of an error in line 19.

31.

Given:

```
10. class One {
11.   public One foo() { return this; }
12. }
13. class Two extends One {
14.   public One foo() { return this; }
15. }
16. class Three extends Two {
17.   // insert method here
18. }
```

Which two methods, inserted individually, correctly complete the Three class? (Choose two.)

- A - public void foo() {}
- B - public int foo() { return 3; }
- C - public Two foo() { return this; }
- D - public One foo() { return this; }
- E - public Object foo() { return this; }

*********** Selamat Mengerjakan **********