JAVA means DURGA SOFT

SCJP / OCJP

Question Bank

Chapter 8 : OOPs

DURGA M.Tech

(Sun certified & Realtime Expert)

Ex. IBM Employee

Trained Lakhs of Students
for last 14 years across INDIA

India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

8.OOPs

Q: 01 Click the Task button.

Place the Output Options in the Actual Output Sequence to indicate the output from this code:

```
class Alpha {
  public void foo( String... args )
    { System.out.print("Alpha:foo "); }
  public void bar( String a )
    { System.out.print("Alpha:bar "); }
}

public class Beta extends Alpha {
  public void foo( String a )
    { System.out.print("Beta:foo "); }
  public void bar( String a )
    { System.out.print("Beta:bar "); }
  public static void main( String[] argv ) {
    Alpha a = new Beta();
    Beta  b = (Beta)a;
    a.foo( "test" ); b.foo( "test" );
    a.bar( "test" ); b.bar( "test" );
  }
}
```

Actual Output Sequence

Place here	Place here	Place here	Place here

Output Options

Alpha:foo	Alpha:bar	Beta:foo	Beta:bar

Done

Solution:
Alpha.foo   Beta.bar   Beta.foo   Beta.bar

Q: 02 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.

Answer: D

Q: 03 Click the Exhibit button.
Which statement is true about the classes and interfaces in the exhibit?

```
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.
Answer: C

Q: 04 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.

Answer: A

Q: 05 Click the Task button.

```
Given:
   class A {
     String name = "A";
     String getName() {
       return name;
     }
     String greeting(){
       return "class A";
     }
   }
   class B extends A {
     String name = "B";
     String greeting() {
       return "class B";
     }
   }
   public class Client {
     public static void main( String[] args ) {
       A a = new A();
       A b = new B();
       System.out.println(a.greeting() + " has name " + a.getName());
       System.out.println(b.greeting() + " has name " + b.getName());
     }
   }
   Place the names "A" and "B" in the following output.         Names

       class  Place here   has name   Place here          A          B

       class  Place here   has name   Place here                    Done
```

Solution:
class   A   has name   A
class   B   hasname   A

Q: 06 Which two statements are true about has-a and is-a relationships? (Choose two.)
A. Inheritance represents an is-a relationship.
B. Inheritance represents a has-a relationship.
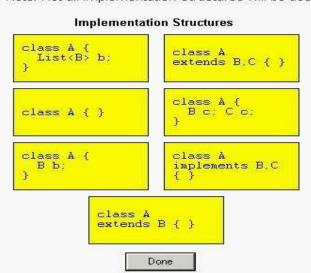C. Interfaces must be used when creating a has-a relationship.

D. Instance variables can be used when creating a has-a relationship.
Answer: A, D

Q: 07 Given:
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.
Answer: C

Q: 08 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); }
Answer: C, D

Q: 09 Click the Task button.

Place the Relations on their corresponding Implementation Structures.
Note: Not all Implementation Structures will be used.

Implementation Structures

```
class A {
   List<B> b;
}
```

```
class A { }
```

```
class A {
   B b;
}
```

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

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

```
class A {
   B c; C c;
}
```

```
class A
implements B,C
{ }
```

Done

Relations

Car is a Vehicle
and
Car is a Collectable

Car has a
SteeringWheel

Car has Wheels

Mini is a Car

Car is an Object

Solution:

Car is a Vehicle		**class A**
and	< ==================>	**implements B , C**
car is a Collectable		**{ }**

Car has a	< ==================>	**class A {**
Steering Wheel		**B b;**
		}

car has Wheels	< ==================>	**class A{**
		List <B> b;
		}
Mini is a Car	< ==================>	**class A**
		extends B{ }
Car is an Object	< ==================>	**class A { }**

Q: 10 Given:
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.

Answer: A

Q: 11 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.

Answer: F

Q: 12 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

Answer: D

Q: 13 Which two statements are true? (Choose two.)
A. An encapsulated, public class promotes re-use.
B. Classes that share the same interface are always tightly encapsulated.
C. An encapsulated class allows subclasses to overload methods, but does NOT allow overriding
methods.
D. An encapsulated class allows a programmer to change an implementation without affecting outside
code.
Answer: A, D

Q: 14 Given:

11. public static void test(String str) {
12. if (str == null | str.length() == 0) {
13. System.out.println("String is empty");
14. } else {
15. System.out.println("String is not empty");
16. }
17. }
And the invocation:
31. test(null);
What is the result?
A. An exception is thrown at runtime.
B. "String is empty" is printed to output.
C. Compilation fails because of an error in line 12.
D. "String is not empty" is printed to output.

Answer: A

Q: 15 Given:
12. public class Wow {
13. public static void go(short n) {System.out.println("short");}
14. public static void go(Short n) {System.out.println("SHORT");}
15. public static void go(Long n) {System.out.println(" LONG");}
16. public static void main(String [] args) {
17. Short y = 6;
18. int z = 7;
19. go(y);
20. go(z);
21. }
22. }
What is the result?
A. short LONG
B. SHORT LONG
C. Compilation fails.
D. An exception is thrown at runtime.

Answer: C

Q: 16 Click the Exhibit button.

Which three code fragments, added individually at line 29, produce the output 100? (Choose three.)

```
10. class Inner {
11.    private int x;
12.    public void setX( int x ) { this.x = x;
}
13.    public int getX() { return x; }
14. }
15.
16. class Outer {
17.    private Inner y;
18.    public void setY( Inner y ) { this.y =
y; }
19.    public Inner getY() { return y; }
20. }
21.
22. public class Gamma {
23.    public static void main( String[] args
) {
24.       Outer o = new Outer();
25.       Inner i = new Inner();
26.       int n = 10;
27.       i.setX( n );
28.       o.setY( i );
29.       // insert code here
30.    System.out.println( o.getY().getX() );
31.    }
32. }
```

A. n = 100;
B. i.setX(100);
C. o.getY().setX(100);
D. i = new Inner(); i.setX(100);
E. o.setY(i); i = new Inner(); i.setX(100);
F. i = new Inner(); i.setX(100); o.setY(i);

Answer: B, C, F

Q: 17 Given:
10. class One {
11. public One() { System.out.print(1); }
12. }
13. class Two extends One {
14. public Two() { System.out.print(2); }

15. }
16. class Three extends Two {
17. public Three() { System.out.print(3); }
18. }
19. public class Numbers{
20. public static void main(String[] argv) { new Three(); }
21. }
What is the result when this code is executed?
A. 1
B. 3
C. 123
D. 321
E. The code runs with no output.

Answer: C

Q: 18 Click the Exhibit button.
What is the outcome of the code?

```
1.  public class Item {
2.     private String desc;
3.     public String getDescription() { return
desc; }
4.     public void setDescription(String d) {
desc = d; }
5.
6.     public static void modifyDesc(Item
item, String desc) {
7.        item = new Item();
8.        item.setDescription(desc);
9.     }
10.    public static void main(String[] args)
{
11.       Item it = new Item();
12.       it.setDescription("Gobstopper");
13.       Item it2 = new Item();
14.       it2.setDescription("Fizzylifting");
15.       modifyDesc(it,
"Scrumdiddlyumptious");
16.
System.out.println(it.getDescription());
17.
System.out.println(it2.getDescription());
18.    }
19. }
```

A. Compilation fails.
B. Gobstopper
Fizzylifting
C. Gobstopper
Scrumdiddlyumptious
D. Scrumdiddlyumptious
Fizzylifting
E. Scrumdiddlyumptious

Answer: B

Q: 19 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; }

Answer: C, D

Q: 20 Click the Exhibit button.

What is the output of the program shown in the exhibit?

```
10.  class Foo {
11.    private int x;
12.    public Foo( int x ) { this.x = x; }
13.    public void setX( int x ) { this.x = x;
}
14.    public int getX() { return x; }
15.  }
16.
17.  public class Gamma {
18.
19.     static Foo fooBar( Foo foo ) {
20.       foo = new Foo( 100 );
21.       return foo;
22.     }
23.
24.     public static void main( String[] args
) {
25.      Foo foo = new Foo( 300 );
26.      System.out.print( foo.getX() + "-" );
27.
28.      Foo fooFoo = fooBar( foo );
29.      System.out.print( foo.getX() + "-" );
30.      System.out.print( fooFoo.getX() + "-"
);
31.
32.      foo = fooBar( fooFoo );
33.      System.out.print( foo.getX() + "-" );
34.      System.out.print( fooFoo.getX() );
35.     }
36.  }
```

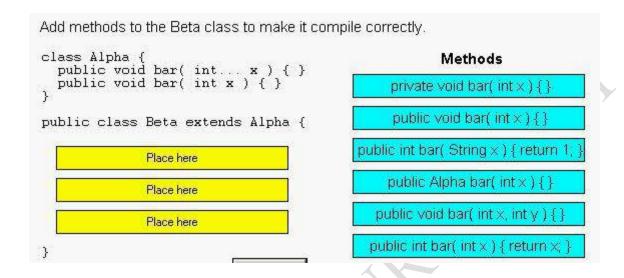A. 300-100-100-100-100
B. 300-300-100-100-100
C. 300-300-300-100-100
D. 300-300-300-300-100

Answer: B

Q: 21 Click the Task button.

Add methods to the Beta class to make it compile correctly.

```
class Alpha {
  public void bar( int... x ) { }
  public void bar( int x ) { }
}

public class Beta extends Alpha {
```

Methods

```
private void bar( int x ){}
public void bar( int x ){}
public int bar( String x ) { return 1; }
public Alpha bar( int x ) { }
public void bar( int x, int y ) {}
public int bar( int x ) { return x; }
```

Place here

Place here

Place here

```
}
```

Solution:

public void bar(int x){ }
public int bar(String x){ return 1; }
public void bar(int x,int y) { }

Q: 22 Given:
10. interface A { public int getValue(); }
11. class B implements A {
12. public int getValue() { return 1; }
13. }
14. class C extends B {
15. // insert code here
16. }
Which three code fragments, inserted individually at line 15, make use of polymorphism?
(Choose three.)
A. public void add(C c) { c.getValue(); }
B. public void add(B b) { b.getValue(); }
C. public void add(A a) { a.getValue(); }
D. public void add(A a, B b) { a.getValue(); }
E. public void add(C c1, C c2) { c1.getValue(); }

Answer: B, C, D

Q: 23 Click the Exhibit button.
What is the result?

```
11.  public class Bootchy {
12.     int bootch;
13.     String snootch;
14.
15.     public Bootchy() {
16.        this("snootchy");
17.        System.out.print("first ");
18.     }
19.
20.     public Bootchy(String snootch) {
21.        this(420, "snootchy");
22.        System.out.print("second ");
23.     }
24.
25.     public Bootchy(int bootch, String
snootch) {
26.        this.bootch = bootch;
27.        this.snootch = snootch;
28.        System.out.print("third ");
29.     }
30.
31.     public static void main(String[] args)
{
32.        Bootchy b = new Bootchy();
33.        System.out.print(b.snootch + " " +
b.bootch);
34.     }
35. }
```

A. snootchy 420 third second first
B. snootchy 420 first second third
C. first second third snootchy 420
D. third second first snootchy 420
E. third first second snootchy 420
F. first second first third snootchy 420

Answer: D

Q: 24 Given:
10. abstract class A {
11. abstract void a1();
12. void a2() { }
13. }
14. class B extends A {
15. void a1() { }
16. void a2() { }
17. }

18. class C extends B { void c1() { } }
and:
A x = new B(); C y = new C(); A z = new C();
What are four valid examples of polymorphic method calls? (Choose four.)
A. x.a2();
B. z.a2();
C. z.c1();
D. z.a1();
E. y.c1();
F. x.a1();
Answer: A, B, D, F

Q: 25 Click the Exhibit button.

What is the result?

```
1.  public class SimpleCalc {
2.     public int value;
3.     public void calculate() { value += 7; }
4.  }

And:

1.  public class MultiCalc extends
SimpleCalc{
2.     public void calculate() { value -= 3; }
3.     public void calculate(int multiplier) {
4.       calculate();
5.       super.calculate();
6.       value *= multiplier;
7.     }
8.     public static void main(String[] args)
{
9.       MultiCalc calculator = new
MultiCalc();
10.      calculator.calculate(2);
11.      System.out.println("Value is: " +
calculator.value);
12.    }
13. }
```

A. Value is: 8
B. Compilation fails.
C. Value is: 12
D. Value is: -12
E. The code runs with no output.
F. An exception is thrown at runtime.
Answer: A

Q: 26 Given:
20. public class CreditCard {
21.
22. private String cardID;
23. private Integer limit;
24. public String ownerName;
25.

26. public void setCardInformation(String cardID,
27. String ownerName,
28. Integer limit) {
29. this.cardID = cardID;
30. this.ownerName = ownerName;
31. this.limit = limit;
32. }
33. }
Which statement is true?
A. The class is fully encapsulated.
B. The code demonstrates polymorphism.
C. The ownerName variable breaks encapsulation.
D. The cardID and limit variables break polymorphism.
E. The setCardInformation method breaks encapsulation.
Answer: C

Q: 27Given:
11. class Animal { public String noise() { return "peep"; } }
12. class Dog extends Animal {
13. public String noise() { return "bark"; }
14. }
15. class Cat extends Animal {
16. public String noise() { return "meow"; }
17. }
...
30. Animal animal = new Dog();
31. Cat cat = (Cat)animal;
32. System.out.println(cat.noise());
What is the result?
A. peep
B. bark
C. meow
D. Compilation fails.
E. An exception is thrown at runtime.
Answer: E
Q: 28 Click the Task button.

Place the Types in one of the Type columns, and the Relationships in the
Relationship column, to define appropriate has-a and is-a relationships.

Type	Relationship	Type	Relationships	Types
Place here	Place here	Animal	is-a	Dog
Forest	Place here	Place here	has-a	Side
Rectangle	Place here	Place here		Tail
Place here	Place here	Programming Book		Square
				Tree
				Book
				Java Book
	Done			Pen

Solution:

Dog	**is-a**	**Animal**
Forest	**has-a**	**Tree**
Rectangle	**has-a**	**Side**
JavaBook	**is-a**	**ProgrammingBook**

Q: 29 Which three statements are true? (Choose three.)

A. A final method in class X can be abstract if and only if X is abstract.

B. A protected method in class X can be overridden by any subclass of X.

C. A private static method can be called only within other static methods in class X.

D. A non-static public final method in class X can be overridden in any subclass of X.

E. A public static method in class X can be called by a subclass of X without explicitly referencing the class X.

F. A method with the same signature as a private final method in class X can be implemented in a subclass of X.

G. A protected method in class X can be overridden by a subclass of A only if the subclass is in the same

package as X.

Answer: B, E, F

Q: 30 Which four statements are true? (Choose four.)

A. Has-a relationships should never be encapsulated.

B. Has-a relationships should be implemented using inheritance.

C. Has-a relationships can be implemented using instance variables.

D. Is-a relationships can be implemented using the extends keyword.

E. Is-a relationships can be implemented using the implements keyword.

F. The relationship between Movie and Actress is an example of an is-a relationship.
G. An array or a collection can be used to implement a one-to-many has-a relationship.
Answer: C, D, E, G

Q: 31Given:
10. interface Jumper { public void jump(); }
...
20. class Animal {}
...
30. class Dog extends Animal {
31. Tail tail;
32. }
...
40. class Beagle extends Dog implements Jumper{
41. public void jump() {} 42. }
...
50. class Cat implements Jumper{
51. public void jump() {}
52. }
Which three are true? (Choose three.)

A. Cat is-a Animal
B. Cat is-a Jumper
C. Dog is-a Animal
D. Dog is-a Jumper
E. Cat has-a Animal
F. Beagle has-a Tail
G. Beagle has-a Jumper

Answer: B, C, F

Q: 32 Click the Exhibit button.
What two must the programmer do to correct the compilation errors? (Choose two.)

```
1.  public class Car {
2.     private int wheelCount;
3.     private String vin;
4.     public Car(String vin) {
5.        this.vin = vin;
6.        this.wheelCount = 4;
7.     }
8.     public String drive() {
9.        return "zoom-zoom";
10.    }
11.    public String getInfo() {
12.       return "VIN: " + vin + " wheels: " +
wheelCount;
13.    }
14. }
```

And:

```
1.  public class MeGo extends Car {
2.     public MeGo(String vin) {
3.        this.wheelCount = 3;
4.     }
5.  }
```

A. insert a call to this() in the Car constructor

B. insert a call to this() in the MeGo constructor

C. insert a call to super() in the MeGo constructor

D. insert a call to super(vin) in the MeGo constructor

E. change the wheelCount variable in Car to protected

F. change line 3 in the MeGo class to super.wheelCount = 3;

Answer: D, E

Q: 33 Given:
10. public class SuperCalc {
11. protected static int multiply(int a, int b) { return a * b;}
12. }
and:
20. public class SubCalc extends SuperCalc{
21. public static int multiply(int a, int b) {
22. int c = super.multiply(a, b);
23. return c;
24. }
25. }
and:
30. SubCalc sc = new SubCalc ();
31. System.out.println(sc.multiply(3,4));
32. System.out.println(SubCalc.multiply(2,2));
What is the result?
A. 12
4
B. The code runs with no output.
C. An exception is thrown at runtime.

D. Compilation fails because of an error in line 21.

E. Compilation fails because of an error in line 22.

F. Compilation fails because of an error in line 31.

Answer: E

Q: 34 Given:

1.public class Blip {

2. protected int blipvert(int x) { return 0; }

3. }

4. class Vert extends Blip {

5. // insert code here

6. }

Which five methods, inserted independently at line 5, will compile? (Choose five.)

A. public int blipvert(int x) { return 0; }

B. private int blipvert(int x) { return 0; }

C. private int blipvert(long x) { return 0; }

D. protected long blipvert(int x) { return 0; }

E. protected int blipvert(long x) { return 0; }

F. protected long blipvert(long x) { return 0; }

G. protected long blipvert(int x, int y) { return 0; }

Answer: A, C, E, F, G

Q: 35 Given:

1. class Pizza {

2. java.util.ArrayList toppings;

3. public final void addTopping(String topping) {

4. toppings.add(topping);

5. }

6. }

7. public class PepperoniPizza extends Pizza {

8. public void addTopping(String topping) {

9. System.out.println("Cannot add Toppings");

10. }

11. public static void main(String[] args) {

12. Pizza pizza = new PepperoniPizza();

13. pizza.addTopping("Mushrooms");

14. }

15. }

What is the result?

```
Given:

10. public class Pizza {
11.    ArrayList toppings;
12.
13.    public final void addTopping(String
topping) {
14.       toppings.add(topping);
15.    }
16.
17.    public void removeTopping(String
topping) {
18.       toppings.remove(topping);
19.    }
20. }

And:

30. class PepperoniPizza extends Pizza {
31.    public void addTopping(String topping) {
32.       System.out.println("Cannot add
Toppings");
33.    }
34.
35.    public void removeTopping(String
topping) {
36.       System.out.println("Cannot remove
Pepperoni");
37.    }
38. }

And:

50. Pizza pizza = new PepperoniPizza();
51. pizza.addTopping("Mushrooms");
52. pizza.removeTopping("Pepperoni");
```

A. Compilation fails.
B. Cannot add Toppings
C. The code runs with no output.
D. A NullPointerException is thrown in Line 4.
Answer: A

Q: 36 Given:
1. interface DoStuff2 {
2. float getRange(int low, int high); }
3.
4. interface DoMore {
5. float getAvg(int a, int b, int c); }
6.
7. abstract class DoAbstract implements DoStuff2, DoMore { }
8.
9. class DoStuff implements DoStuff2 {
10. public float getRange(int x, int y) { return 3.14f; } }
11.
12. interface DoAll extends DoMore {
13. float getAvg(int a, int b, int c, int d); }
What is the result?

A. The file will compile without error.

B. Compilation fails. Only line 7 contains an error.

C. Compilation fails. Only line 12 contains an error.

D. Compilation fails. Only line 13 contains an error.

E. Compilation fails. Only lines 7 and 12 contain errors.

F. Compilation fails. Only lines 7 and 13 contain errors.

G. Compilation fails. Lines 7, 12, and 13 contain errors.

Answer: A

Q: 37 Click the Exhibit button.
Given:
25. A a = new A();
26. System.out.println(a.doit(4, 5));
What is the result?

```
1.  public class A {
2.     public String doit(int x, int y) {
3.        return "a";
4.     }
5.
6.     public String doit(int... vals) {
7.        return "b";
8.     }
9.  }
```

A. Line 26 prints "a" to System.out.

B. Line 26 prints "b" to System.out.

C. An exception is thrown at line 26 at runtime.

D. Compilation of class A will fail due to an error in line 6.

Answer: A

Q: 38 Given:
1. class TestA {
2. public void start() { System.out.println("TestA"); }
3. }
4. public class TestB extends TestA {
5. public void start() { System.out.println("TestB"); }
6. public static void main(String[] args) {
7. ((TestA)new TestB()).start();
8. }
9. }
What is the result?
A. TestA
B. TestB
C. Compilation fails.
D. An exception is thrown at runtime.
Answer: B

Q: 39 Given:
1. interface TestA { String toString(); }
2. public class Test {
3. public static void main(String[] args) {

4. System.out.println(new TestA() {
5. public String toString() { return "test"; }
6. });
7. }
8. }
What is the result?
A. test
B. null
C. An exception is thrown at runtime.
D. Compilation fails because of an error in line 1.
E. Compilation fails because of an error in line 4.
F. Compilation fails because of an error in line 5.

Answer: A

Q: 40 Given:
11. public class ItemTest {
12. private final int id;
13. public ItemTest(int id) { this.id = id; }
14. public void updateId(int newId) { id = newId; }
15.
16. public static void main(String[] args) {
17. ItemTest fa = new ItemTest(42);
18. fa.updateId(69);
19. System.out.println(fa.id);
20. }
21. }
What is the result?
A. Compilation fails.
B. An exception is thrown at runtime.
C. The attribute id in the Item object remains unchanged.
D. The attribute id in the Item object is modified to the new value.
E. A new Item object is created with the preferred value in the id attribute.
Answer: A

Q: 41 Given:
10. class One {
11. void foo() { }
12. }
13. class Two extends One {
14. //insert method here
15. }
Which three methods, inserted individually at line 14, will correctly complete class Two? (Choose three.)
A. int foo() { /* more code here */ }
B. void foo() { /* more code here */ }
C. public void foo() { /* more code here */ }
D. private void foo() { /* more code here */ }
E. protected void foo() { /* more code here */ }

Answer: B, C, E

Q: 42 Given:
10. interface Data { public void load(); }
11. abstract class Info { public abstract void load(); }
Which class correctly uses the Data interface and Info class?
A. public class Employee extends Info implements Data {
public void load() { /*do something*/ }
}
B. public class Employee implements Info extends Data {
public void load() { /*do something*/ } }
C. public class Employee extends Info implements Data
public void load(){ /*do something*/ }
public void Info.load(){ /*do something*/ }
}
D. public class Employee implements Info extends Data {
public void Data.load(){ /*do something*/ }
public void load(){ /*do something*/ }
}
E. public class Employee implements Info extends Data {
public void load(){ /*do something*/ }
public void Info.load(){ /*do something*/ }
}
F. public class Employee extends Info implements Data{
public void Data.load() { /*do something*/ }
public void Info.load() { /*do something*/ } }
Answer: A

Q: 43 Given:
11. public abstract class Shape {
12. int x;
13. int y;
14. public abstract void draw();
15. public void setAnchor(int x, int y) {
16. this.x = x;
17. this.y = y;
18. }
19. }
and a class Circle that extends and fully implements the Shape class.
Which is correct?
A. Shape s = new Shape();
s.setAnchor(10,10);
s.draw();
B. Circle c = new Shape();
c.setAnchor(10,10);
c.draw();
C. Shape s = new Circle();
s.setAnchor(10,10);
s.draw();
D. Shape s = new Circle();

```
s->setAnchor(10,10);
s->draw();
E. Circle c = new Circle();
c.Shape.setAnchor(10,10);
c.Shape.draw();
```

Answer: C

Q: 44 Click the Exhibit button.
What is the result?

```
11. class Person {
12.    String name = "No name";
13.    public Person(String nm) { name = nm; }
14. }
15.
16. class Employee extends Person {
17.    String empID = "0000";
18.    public Employee(String id) { empID =
id; }
19. }
20.
21. public class EmployeeTest {
22.    public static void main(String[] args)
{
23.       Employee e = new Employee("4321");
24.       System.out.println(e.empID);
25.    }
26. }
```

A. 4321
B. 0000
C. An exception is thrown at runtime.
D. Compilation fails because of an error in line 18.
Answer: D

Q: 45 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;
Answer: B

Q: 46 Given:
1. class SuperClass {
2. public A getA() {
3. return new A();
4. }
5. }
6. class SubClass extends SuperClass {
7. public B getA(){
8. return new B();
9. }
10. }
Which statement is true?
A. Compilation will succeed if A extends B.
B. Compilation will succeed if B extends A.
C. Compilation will always fail because of an error in line 7.
D. Compilation will always fail because of an error in line 8.

Answer: B

Q: 47 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. new B().process();
19. }
What is the result?
A. B
B. The code runs with no output.
C. Compilation fails because of an error in line 12.
D. Compilation fails because of an error in line 15.
E. Compilation fails because of an error in line 18.
Answer: A

Q: 48 Given:
11. class ClassA {}
12. class ClassB extends ClassA {}
13. class ClassC extends ClassA {}

and:
21. ClassA p0 = new ClassA();
22. ClassB p1 = new ClassB();
23. ClassC p2 = new ClassC();
24. ClassA p3 = new ClassB();
25. ClassA p4 = new ClassC();
Which three are valid? (Choose three.)
A. p0 = p1;
B. p1 = p2;
C. p2 = p4;
D. p2 = (ClassC)p1;
E. p1 = (ClassB)p3;
F. p2 = (ClassC)p4;
Answer: A, E, F

Q: 49 Given:
11. abstract class Vehicle { public int speed() { return 0; }
12. class Car extends Vehicle { public int speed() { return 60; }
13. class RaceCar extends Car { public int speed() { return 150; }
...
21. RaceCar racer = new RaceCar();
22. Car car = new RaceCar();
23. Vehicle vehicle = new RaceCar();
24. System.out.println(racer.speed() + ", " + car.speed()
25. + ", " + vehicle.speed());
What is the result?
A. 0, 0, 0
B. 150, 60, 0
C. Compilation fails.
D. 150, 150, 150
E. An exception is thrown at runtime.

Answer: D

Q: 50 Given code in separate source files:

10. public class Foo {
11. public int a;
12. public Foo() { a = 3; }
13. public void addFive() { a += 5;}
14. } and: 20. public class Bar extends Foo {
21. public int a;
22. public Bar() { a = 8; }
23. public void addFive() { this.a += 5; }
24. } invoked with:
30. Foo foo = new Bar();
31. foo.addFive();
32. System.out.println("Value: " + foo.a);
What is the result?

A. Value: 3
B. Value: 8
C. Value: 13
D. Compilation fails.
E. The code runs with no output.
F. An exception is thrown at runtime.
Answer: A

Q: 51 Which Man class properly represents the relationship "Man has a best friend who is a Dog"?
A. class Man extends Dog { }
B. class Man implements Dog { }
C. class Man { private BestFriend dog; }
D. class Man { private Dog bestFriend; }
E. class Man { private Dog<bestFriend>; }
F. class Man { private BestFriend<dog>; }
Answer: D

Q: 52 Given:
10. interface Foo { int bar(); }
11. public class Sprite {
12. public int fubar(Foo foo) { return foo.bar(); }
13. public void testFoo() {
14. fubar(
15. // insert code here
16.);
17. }
18. }

Which code, inserted at line 15, allows the class Sprite to compile?
A. Foo { public int bar() { return 1; }
B. new Foo { public int bar() { return 1; }
C. new Foo() { public int bar() { return 1; }
D. new class Foo { public int bar() { return 1; }
Answer: C

Q: 53 Click the Task button.

```
Given:  public class Doubler {
            public static int doubleMe( Holder h) {
              return h.getAmount() * 2;
            }
        }
and:    public class Holder {
            int amount = 10;
            public void doubleAmount(){ amount = Doubler.doubleMe( this );}
            public int getAmount(){ return amount;}
            //more code here
        }
```

Place the code fragments in position to reduce the coupling between Doubler and Holder.

```
public class Doubler {
  public static int doubleMe( [Place here]      h) {
    return [Place here]  * 2;
  }
}
```

```
public class Holder {
  int amount = 10;
  public void doubleAmount(){ amount = Doubler.doubleMe( [Place here]  );}
  public int getAmount(){ return amount;}
  //more code here
}
```

Code Fragments

void	Holder	int	Doubler
h.getAmount()	h	this	amount

[Done]

Solution:

1. int   2. h   3. amount.

Q: 54 Given classes defined in two different files:

1. package packageA;
2. public class Message {
3. String getText() { return "text"; }
4. }
and:
1. package packageB;
2. public class XMLMessage extends packageA.Message {
3. String getText() { return "<msg>text</msg>";}
4. public static void main(String[] args) {
5. System.out.println(new XMLMessage().getText());
6. }
7. }
What is the result of executing XMLMessage.main?
A. text
B. An exception is thrown at runtime.
C. Compilation fails because of an error in line 2 of XMLMessage.
D. Compilation fails because of an error in line 3 of XMLMessage.

Answer: D

Q: 55 Given:

1. public class A {
2. public void doit() {
3. }
4. public String doit() {
5. return "a";
6. }
7. public double doit(int x) {
8. return 1.0;
9. }
10. }
What is the result?
A. An exception is thrown at runtime.
B. Compilation fails because of an error in line 7.
C. Compilation fails because of an error in line 4.
D. Compilation succeeds and no runtime errors with class A occur.
Answer: C

Question: 56
Click the Exhibit button.
1. public class GoTest {
2. public static void main(String[] args) {
3. Sente a = new Sente(); a.go();
4. Goban b = new Goban(); b.go();
5. **Stone c = new Stone(); c.go();**
6. **}**
7. **}**
8.
9. **class Sente implements Go {**
10. public void go() { System.out.println("go in Sente."); }
11. }
12.
13. class Goban extends Sente {
14. public void go() { System.out.println("go in Goban"); }
15. **}**
16.
17. class Stone extends Goban implements Go { }
18.
19. interface Go { public void go(); }
What is the result?
A. go in Goban
go in Sente
go in Sente
B. go in Sente
go in Sente
go in Goban

C. go in Sente
go in Goban
go in Goban
D. go in Goban
go in Goban
go in Sente
E. Compilation fails because of an error in line 17.

Answer: C

Question: 57
Click the Exhibit button.

1. public class Employee {
2. String name;
3. double baseSalary;
4. Employee(String name, double baseSalary) {
5. this.name = name;
6. this.baseSalary = baseSalary;
7. }
8. }
And:
1. public class Salesperson extends Employee {
2. double commission;
3. public Salesperson(String name, double baseSalary,
4. double commission) {
5. // insert code here
6. }
7. }

Which code, inserted at line 7, completes the Salesperson constructor?
A. this.commission = commission;
B. superb();
commission = commission;
C. this.commission = commission;
superb();
D. super(name, baseSalary);
this.commission = commission;
E. super();
this.commission = commission;
F. this.commission = commission;
super(name, baseSalary);

Answer: D

Question: 58
Given:
1. public class Team extends java.util.LinkedList {
2. public void addPlayer(Player p) {
3. add(p);

4. }
5. **public void compete(Team opponent) { /* more code here */ }**
6. }
7. class Player { /* more code here */ }

Which two are true? (Choose two.)
A. This code will compile.
B. This code demonstrates proper design of an is-a relationship.
C. This code demonstrates proper design of a has-a relationship.
D. A Java programmer using the Team class could remove Player
objects from a Team object.
Answer: AD

Question: 59
Given:
5. class Atom {
6. Atom() { System.out.print("atom "); }
7. }
8. class Rock extends Atom {
9. Rock(String type) { System.out.print(type); }
10. }
11. public class Mountain extends Rock {
12. Mountain() {
13. super("granite ");
14. new Rock("granite ");
15. }
16. public static void main(String[] a) { new Mountain(); }
17. }
What is the result?
A. Compilation fails.
B. atom granite
C. granite granite
D. atom granite granite
E. An exception is thrown at runtime.
F. atom granite atom granite
Answer: F

Question: 60
Given:
5. class Building { }
6. public class Barn extends Building {
7. public static void main(String[] args) {
8. Building build1 = new Building();
9. Barn barn1 = new Barn();
10. Barn barn2 = (Barn) build1;
11. Object obj1 = (Object) build1;
12. String str1 = (String) build1;
13. Building build2 = (Building) barn1;
14. }
15. }

Which is true?

A. If line 10 is removed, the compilation succeeds.

B. If line 11 is removed, the compilation succeeds.

C. If line 12 is removed, the compilation succeeds.

D. If line 13 is removed, the compilation succeeds.

E. More than one line must be removed for compilation to succeed.

Answer: C

Question: 61

A company has a business application that provides its users with many different reports receivables reports, payables reports, revenue projects, and so on. The company has just urchased some new, state-of-the-art, wireless printers, and a programmer has been assigned the task of enhancing all of the reports to use not only the company's old printers, but the new wireless printers as well. When the programmer starts looking into the application, the programmer discovers that because of the design of the application, it is necessary to make changes to each report to support the new printers.Which two design concepts most likely explain this situation? (Choose two.)

A. Inheritance

B. Low cohesion

C. Tight coupling

D. High cohesion

E. Loose coupling

F. Object immutability

Answer: B,C

Question: 62
Given:
31. class Foo {
32. public int a = 3;
33. public void addFive() { a += 5; System.out.print("f "); }
34. }
35. class Bar extends Foo {
36. public int a = 8;
37. public void addFive() { this.a += 5; System.out.print("b "); }
38. }
Invoked with:
Foo f = new Bar();

f.addFive();
System.out.println(f.a);
What is the result?
A. b 3
B. b 8
C. b 13
D. f 3
E. f 8
F. f 13
G. Compilation fails.
H. An exception is thrown at runtime.
Answer: A

Question: 63
A company that makes Computer Assisted Design (CAD) software has, within its application, some utility classes that are used to perform 3D rendering tasks. The company's chief scientist has just improved the performance of one of the utility classes' key rendering algorithms, and has assigned a programmer to replace the old algorithm with the new algorithm. When the programmer begins researching the utility classes, she is happy to discover that the algorithm to be replaced exists in only one class. The programmer reviews that class's API, and replaces the old algorithm with the new algorithm, being careful that her changes adhere strictly to the class's API. Once testing has begun, the programmer discovers that other classes that use the class she changed are no longer working properly. What design flaw is most likely the cause of these new bugs?
A. Inheritance
B. Tight coupling
C. Low cohesion
D. High cohesion
E. Loose coupling
F. Object immutability
Answer: B

Question: 64
Given:
5. class Thingy { Meter m = new Meter(); }
6. class Component { void go() { System.out.print("c"); } }
7. class Meter extends Component { void go() { System.out.print("m"); } }
8.
9. class DeluxeThingy extends Thingy {
10. public static void main(String[] args) {
11. DeluxeThingy dt = new DeluxeThingy();
12. dt.m.go();
13. Thingy t = new DeluxeThingy();
14. t.m.go();
15. }
16. }
Which two are true? (Choose two.)
A. The output is mm.

B. The output is mc.
C. Component is-a Meter.
D. Component has-a Meter.
E. DeluxeThingy is-a Component.
F. DeluxeThingy has-a Component.
Answer: A,F

Question: 65
Given:
1. class X {
2. X() { System.out.print(1); }
3. X(int x) {
4. this(); System.out.print(2);
5. }
6. }
7. public class Y extends X {
8. Y() { super(6); System.out.print(3); }
9. Y(int y) {
10. this(); System.out.println(4);
11. }
12. public static void main(String[] a) { new Y(5); }
13. }
What is the result?
A. 13
B. 134
C. 1234
D. 2134
E. 2143
F. 4321
Answer: C

Question: 66
Given:
2. public class Hi {
3. void m1() { }
4. protected void() m2 { }
5. }
6. class Lois extends Hi {
7. // insert code here
8. }
Which four code fragments, inserted independently at line 7, will compile? (Choose four.)
A. public void m1() { }
B. protected void m1() { }
C. private void m1() { }

D. void m2() { }
E. public void m2() { }
F. protected void m2() { }
G. private void m2() { }
Answer: A,B,E,F

Question: 67
Given:
11. public interface A { public void m1(); }
12.
13. class B implements A { }
14. class C implements A { public void m1() { } }
15. class D implements A { public void m1(int x) { } }
16. abstract class E implements A { }
17. abstract class F implements A { public void m1() { } }
18. abstract class G implements A { public void m1(int x) { } }
What is the result?
A. Compilation succeeds.
B. Exactly one class does NOT compile.
C. Exactly two classes do NOT compile.
D. Exactly four classes do NOT compile.
E. Exactly three classes do NOT compile.
Answer: C

Question: 68
Given:
3. interface Fish { }
4. class Perch implements Fish { }
5. class Walleye extends Perch { }
6. class Bluegill { }
7. public class Fisherman {
8. public static void main(String[] args) {
9. Fish f = new Walleye();
10. Walleye w = new Walleye();
11. Bluegill b = new Bluegill();
12. if(f instanceof Perch) System.out.print("f-p ");
13. if(w instanceof Fish) System.out.print("w-f ");
14. if(b instanceof Fish) System.out.print("b-f ");
15. }
16. }
What is the result?
A. w-f
B. f-p w-f
C. w-f b-f
D. f-p w-f b-f
E. Compilation fails.
F. An exception is thrown at runtime.
Answer: B

Question: 69
11. public interface A111 {
12. String s = "yo";
13. public void method1();
14. }
17. interface B { }
20. interface C extends A111, B {
21. public void method1();
22. public void method1(int x);
23. } What is the result?
A. Compilation succeeds.
B. Compilation fails due to multiple errors.
C. Compilation fails due to an error only on line 20.
D. Compilation fails due to an error only on line 21.
E. Compilation fails due to an error only on line 22.
F. Compilation fails due to an error only on line 12.
Answer: A

Question: 70
10. import java.io.*;
11. class Animal {
12. Animal() { System.out.print("a"); }
13. }
14. class Dog extends Animal implements Serializable {
15. Dog() { System.out.print("d"); }
16. }
17. public class Beagle extends Dog { }
If an instance of class Beagle is created, then Serialized, then deSerialized, what is the result?
A. ad
B. ada
C. add
D. adad
E. Compilation fails.
F. An exception is thrown at runtime.
Answer: B

Question: 71
A team of programmers is involved in reviewing a proposed design for a new utility class. After some discussion, they realize that the current design allows other classes to access methods in the utility class that should be accessible only to methods within the utility class itself. What design issue has the team discovered?
A. Tight coupling
B. Low cohesion
C. High cohesion
D. Loose coupling
E. Weak encapsulation

F. Strong encapsulation
Answer: E

Question: 72
Given that: Gadget has-a Sprocket and Gadget has-a Spring and Gadget is-a Widget and Widget has-a Sprocket Which two code fragments represent these relationships? (Choose two.)
A. class Widget { Sprocket s; } class Gadget extends Widget { Spring s; }
B. class Widget { } class Gadget extends Widget { Spring s1; Sprocket s2; }
C. class Widget { Sprocket s1; Spring s2; } class Gadget extends Widget { }
D. class Gadget { Spring s; } class Widget extends Gadget{ Sprocket s; }
E. class Gadget { } class Widget extends Gadget{ Sprocket s1; Spring s2; }
F. class Gadget { Spring s1; Sprocket s2; } class Widget extends Gadget{ }
Answer: A,C

Question: 73
11. class Alpha {
12. public void foo() { System.out.print("Afoo "); }
13. }
14. public class Beta extends Alpha {
15. public void foo() { System.out.print("Bfoo "); }
16. public static void main(String[] args) {
17. Alpha a = new Beta();
18. Beta b = (Beta)a;
19. a.foo(); 20. b.foo();
21. }
22. }
What is the result?
A. Afoo Afoo
B. Afoo Bfoo
C. Bfoo Afoo
D. Bfoo Bfoo
E. Compilation fails.
F. An exception is thrown at runtime.
Answer: D

Question: 74
Click the Exhibit button.
11. class Payload {
12. private int weight;
13. public Payload(int wt) { weight = wt; }
13. public void setWeight(mt w) { weight = w; }
15. public String toString { return Integer.toString(weight); }
16. }
17.
18. public class TestPayload {
19. static void changePayload(Payload p) {
20. /* insert code here */
21. }

22.
23. public static void main(String[] args) {
24. Payload p = new Payload();
25. p.setWeight(1024);
26. changePayload(p);
27. System.out.println("The value of p is "+ p);
28. }
29. }
Which statement, placed at line 20, causes the code to print "The
value of p is 420."?
A. p.setWeight(420);
B. p.changePayload(420);
C. p = new Payload(420);
D. Payload.setWeight(420);
E. p = Payload.setWeight(420);
F. p = new Payload();
p.setWeight(420);
Answer: A

.