**Question 159 :**

Which four are true? (Choose four.)

A. Has-a relationships should never be encapsulated.-x

B. Has-a relationships should be implemented using inheritance.-x : is a 임.

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

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

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

F. The relationship between Movie and Actress is an example of an is-a relationship.-x

G. An array or a collection can be used to implement a one-to-many has-a relationship.-O

**Question 160 :**

Which two are true about has-a and is-a relationships? (Choose two.)

A. Inheritance represents an is-a relationship. -O

B. Inheritance represents a has-a relationship. - x

C. Interfaces must be used when creating a has-a relationship. –x is a

D. Instance variables can be used when creating a has-a relationship.-O

**Question 161 :**

Given:

10. interface Jumper { public void jump(); }

......

20. class Animal {}

...... (=is a)

30. class Dog extends Animal {

31. Tail tail; // has a

32. }

...... (=is a)

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-O

C. Dog is-a Animal-O

D. Dog is-a Jumper

E. Cat has-a Animal

F. Beagle has-a Tail-O

G. Beagle has-a Jumper

**Question 1 :**

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();// 메모리에 Foo, Alpha, Beta 올라가고, Delta 는 아직 안 올라감

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; // Delta 가 아직 메모리에 올라가지 않아 에러.

C. Foo f= (Alpha)x;

D. Beta b = (Beta)(Alpha)x;

**Question 9 :**

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> }

**Question 18**

Given:

1. public interface A {

2. String DEFAULT\_GREETING = “Hello World”;

3. public void method1();

4. }

A programmer wants to create an interface called B that has A as its parent. Which interface declaration is correct?

A. public interface B extends A { }

B. public interface B implements A {}

C. public interface B instanceOf A {}

D. public interface B inheritsFrom A { }// interface 간 상속도 가능

**Question 19**

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.

**Question 21**

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();// 추상클래스라 new 안됨.

s.setAnchor(10,10);

s.draw();

B. Circle c = new Shape();();// 추상클래스라 new 안됨

c.setAnchor(10,10);

c.draw();

C. Shape s = new Circle();

s.setAnchor(10,10);

s.draw();

D. Circle c = new Circle();

c.Shape.setAnchor(10,10);//c.Shape 문법적으로 안맞음

c.Shape.draw();

**Question 23 :**

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\*/ } // 둘다 load 가 있으므로 하나만 하면 됨.

}

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\*/ }// 문법 오류

}

D. public class Employee implements Info extends Data {

public void Data.load() { /\*d something \*/ }// 문법 오류

public void load() { /\*do something \*/ }

}

E. public class Employee implements Info extends Data {

public void load() { /\*do something \*/ }

public void Info.load(){ /\*do something\*/ }}// 문법 오류

}

F. public class Employee extends Info implements Data{

public void Data.load() { /\*do something\*/ }}// 문법 오류

public void Info.load() { /\*do something\*/ }}// 문법 오류

}

**Question 24 :**

Given:

11. public abstract class Shape {

12. private int x;

13. private 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. }

Which two classes use the Shape class correctly? (Choose two.)

A. public class Circle implements Shape {

private int radius;

}

B. public abstract class Circle extends Shape {

private int radius;

}

C. public class Circle extends Shape {

private int radius;

public void draw();

}

D. public abstract class Circle implements Shape {

private int radius;

public void draw();

}

E. public class Circle extends Shape {

private int radius;

public void draw() {/\* code here \*/}

}

F. public abstract class Circle implements Shape {

private int radius;

public void draw() { / code here \*/ }

}

**Question 38 :**

A programmer is designing a class to encapsulate the information about an inventory item. A JavaBeans component is needed to do this. The Inventory item class has private instance variables to store the item information:

10. private int itemId;

11. private String name;

12. private String description;

Which method signature follows the JavaBeans naming standards for modifying the itemId instance variable?

A. itemID(int itemId)

B. update(int itemId)

C. setItemId(int itemId)

D. mutateItemId(int itemId)

E. updateItemID(int itemId)

**Question 42**

//자식 클래스는 오버라이딩 시, 부모보다 보안을 강화시킬 수 없다.

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 \*/ }

**Question 48 :**

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 rims with no output.

**Question 50 :**

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 is true?

A. The code will compile without changes.-부모에 인수 있는 생성자가 있으므로 에러

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.

**Question 138 :**

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;

**Question 139 :**

Given:

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.

**Question 140 :**

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.

**Question 141 :**

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();

Which four are 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();

**Question 143**

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 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.

**Question 147**

147. 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

**Question 150 :**

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); }

**Question 155 :**

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; }

**Question156**

Given:

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.

**Question 157**

Given:

10. public class SuperCaic {

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.

**Question 171**

Given:

1. public class Person {

2. private String name;

3. public Person(String name) { this.name = name; }

4. public boolean equals(Person p) {

5. return p.name.equals(this.name);

6. }

7. }

Which is true?

A. The equals method does NOT properly override the Object.equals method.

B. Compilation fails because the private attribute p.name cannot be accessed in line 5.

C. To work correctly with hash-based data structures, this class must also implement the hashCode method.

D. When adding Person objects to a java.util.Set collection, the equals method in line 4 will prevent duplicates.

**Question 196**

Given:

11. interface DeclareStuff{

12. public static final int EASY = 3;

13. void doStuff(int t); }

14. public class TestDeclare implements DeclareStuff {

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

16. int x=5;

17. new TestDeclare().doStuff(++x);

18. }

19. void doStuff(int s) {

20. s += EASY + ++s;

21. System.out.println(”s “ + s);

22. }

23. }

What is the result?

A. s 14

B. s 16

C. s 10

D. Compilation fails.

E. An exception is thrown at runtime.

**Question 222 :**

Given:

11. class Cup { }

12. class PoisonCup extends Cup { }

21. public void takeCup(Cup c) {

22. if(c instanceof PoisonCup) {

23. System.out.println(”Inconceivable!”);

24. } else if(c instanceof Cup) {

25. System.out.println(”Dizzying intellect!”);

26. } else {

27. System.exit(0);

28. }

29. }

And the execution of the statements:

Cup cup = new PoisonCup();

takeCup(cup);

What is the output?

A. Inconceivable!

B. Dizzying intellect!

C. The code runs with no output.

D. An exception is thrown at runtime.

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