

Section=>1

-----

1. 

```
enum Element{
    EARTH,WIND,
    FIRE{public String info(){return "Hot";}
};
public String info(){return "element";}
}
```
2. A, C
3. 

```
package com.sun.cert;
import java.util.*;
public class AddressBook{
    ArrayList entries;
}
```
4. C, D
5. F
6. A
7. A
8. D
9. D
10. D
11. D
12. B
13. A
14. A
15. C
16. A
17. 

```
package alpha;
public class Alpha{
    private String alpha;
    public Alpha(){ this("A"); }
    protected Alpha(String a){ alpha=a; }
}
package beta;
public class Beta extends alpha.Alpha{
    private Beta(String a){ super(a); }
}
```
18. D

```

19. public class Single{
        private static Single instance;
        public static Single getInstance(){
            if(instance==null) instance = create();
            return instance;
        }
        protectedSingle() { }
        staticSingle create () { return new Single (); }
    }
    class SingleSub extends Shape{
    }

```

20. D

21. D

```

22. public int update(int quantity,int adjust){
        quantity=quantity+adjust;
        return quantity;
    }
    public void call Update() {
        int quant=100;
        quant=update(quant,320);
        System.out.println("the quantity is " +quant);
    }

```

23. C, E

24. B, F

25. ->C and D

26. ->B

27. C

28. A

29. D and F

---

Section=>2

1. c.
2. a, c, d, e. b
3. a, c, d, f.
4. b
5. b.
6. d
7. c
8. d
9. a
10. h

- 11. c
- 12. f
- 13. b
- 14. a
- 15. d
- 16. d
- 17. f
- 18. d
- 19. b
- 20. a, c
- 21. e.
- 22. b, c, d, e, g, h, i, j, l
- 23. q
- 24. b, c, d, g, h, l
- 25. b, c, d, g, h, l
- 26. b, c, d

Section=>3

-----

- 1. D
- 2. Alpha.foo Beta.ba Beta.foo Beta.bar
- 3. A
- 4. C
- 5. class A has name A  
class B hasname A
- 6. C
- 7. A, D
- 8. C, D
- 9. Car is a vehical                      class A  
    And   =====→ implements B, C  
Car is a collectable                    {}  
  
Car has a       =====→ class A{  
Steering wheel                        B b;  
  
Car has wheel       ====→ class A{ List<B> b; }
- Mini is a car   => class A extends B{}
- Car is an object ==> class A{}
- 10. F
- 11. A
- 12. D

13. A, D

14. A

15. C

16. B, C, F

17. C

18. B

19. C, D

20. B

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

22. B, C, D

23. D

24. A, B, D, F

25. A

26. C

27. E

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

29. B, E, F

30. C, D, E, G

31. B, C, F

32. D, E

33. E

34. A, C, E, F, G

35. A

36. A

37. B

38. A

39. A

40. A

41. B, C, E

42. A

43. C

44. D

45. B

46. B

47.A

48.A, E, F

49.D

50.A

51.D

52.C

53. 1. int      2. h      3. amount.

54.D

55.C

56.C

57.D

58.A, D

59.A

60.B

61.E

62.B

E

63. E

64. E and F

65. A

66. A, C, and D

67. D

68. A, B, E, and F

69. D

70. C

71. A and B

72. C

73. D

---

Section=>4

-----

1. D
2. B
3. D
4. C
5. D
6. B, D
7. D
8. E
9. E
10. E
11. G
12. 

```
class Incr {
    public static void main(String[] args) {
        Integer x = 7;
        int y = 2;
        x *= x;
        y *= y;
        y *= y;
        x -= y;
        System.out.println(x);
    }
}
```

Yeah, we know it's kind of puzzle-y, but you might encounter something like it on the real exam.

13. A, D, F
14. F
15. D
16. C
17. C
18. G
19. F
20. A
21. A
22. D
23. C, E