***Object-oriented programming (OOPs)***

**Ques: 01 Given:**

**public class Plant {**

**private String name;**

**public Plant(String name) { this.name = name; }**

**public String getName() { return name; }**

**}**

**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.

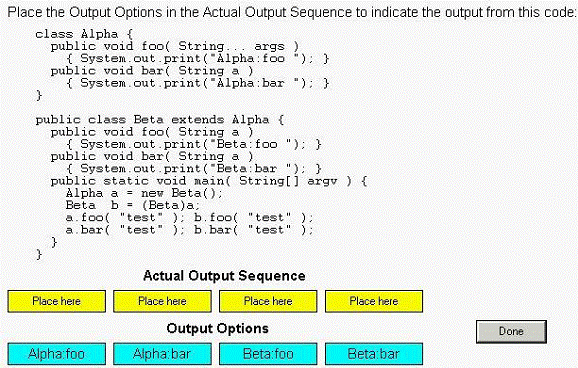
C. The code will compile if public Plant() { Tree(); } is added to the Plant class.

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: B

**Ques: 02 Click the Task button.**



**Ques: 03 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

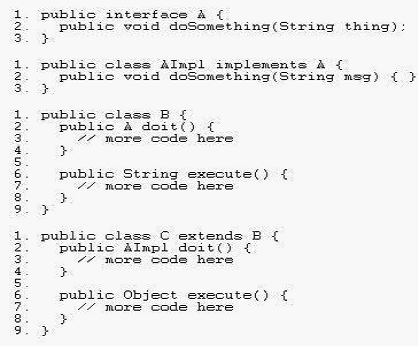
B. Short Long

C. Compilation fails.

D. An exception is thrown at runtime.

**Ques: 04 Click the Exhibit button.**

**Which statement is true about the classes and interfaces in the exhibit?**



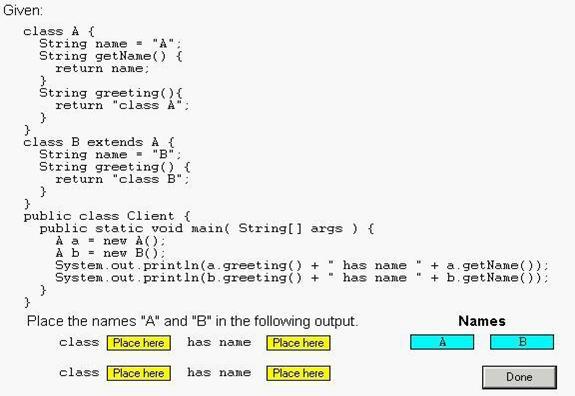
A. Compilation will succeed for all classes and interfaces.

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.

**Ques: 05 Click the Task button.**



**Ques: 06 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.

**Ques: 07 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.

**Ques: 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;

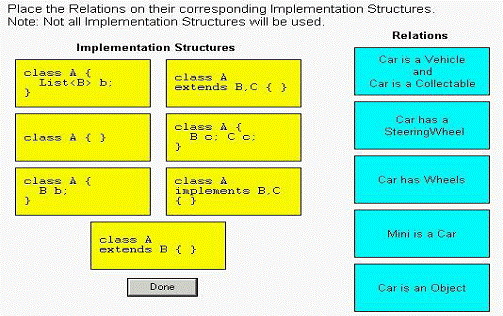
B. Change line 2 to :protected int a;

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); }

**Ques: 09 Click the Task button.**



**Ques: 10 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.

**Ques: 11 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**

**Ques: 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

**Ques: 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.

**Ques: 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.

**Ques: 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

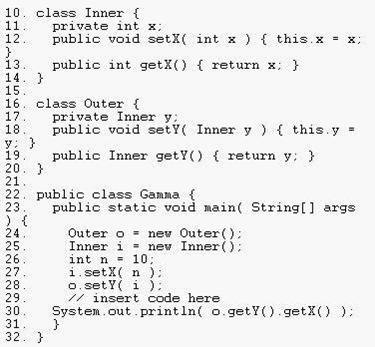
B. SHORT LONG

C. Compilation fails.

D. An exception is thrown at runtime.

**Ques: 16 Click the Exhibit button.**

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



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 );

**Ques: 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

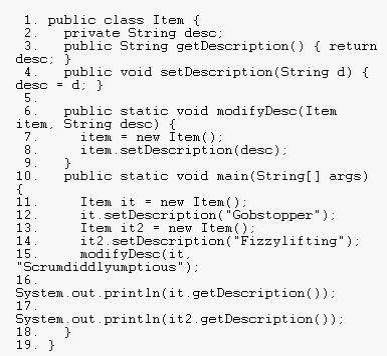
C. 123

D. 321

E. The code runs with no output.

**Ques: 18 Click the Exhibit button.**

**What is the outcome of the code?**



A. Compilation fails.

B. Gobstopper

Fizzylifting

C. Gobstopper

Scrumdiddlyumptious

D. Scrumdiddlyumptious

Fizzylifting

E. Scrumdiddlyumptious

**Ques: 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; }

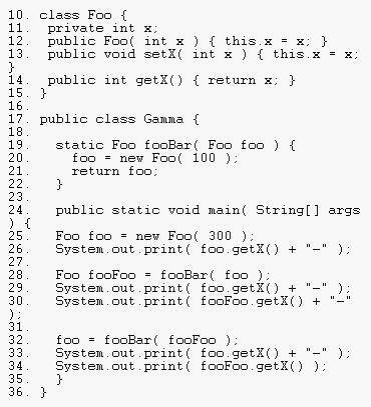
C. public Two foo() { return this; }

D. public One foo() { return this; }

E. public Object foo() { return this; }

**Ques: 20 Click the Exhibit button.**

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



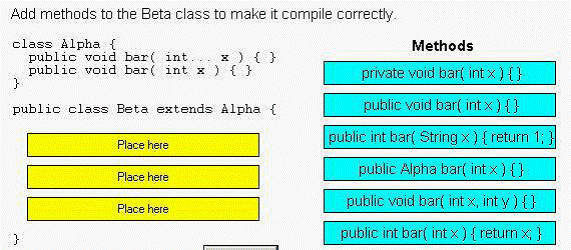
A. 300-100-100-100-100

B. 300-300-100-100-100

C. 300-300-300-100-100

D. 300-300-300-300-100

**Ques: 21 Click the Task button.**



**Solution:**

**public void bar(int x){ }**

**public int bar(String x){ return 1; }**

**public void bar(int x,int y) { }**

**Ques: 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(); }

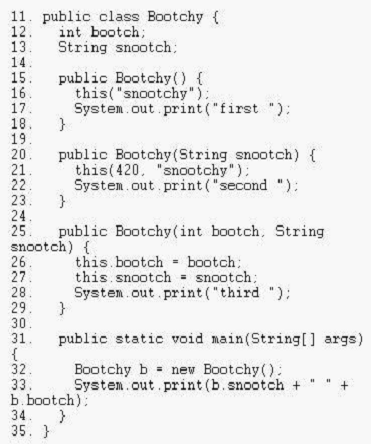
C. public void add(A a) { a.getValue(); }

D. public void add(A a, B b) { a.getValue(); }

E. public void add(C c1, C c2) { c1.getValue(); }

**Ques: 23 Click the Exhibit button.**

**What is the result?**



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

**Ques: 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();

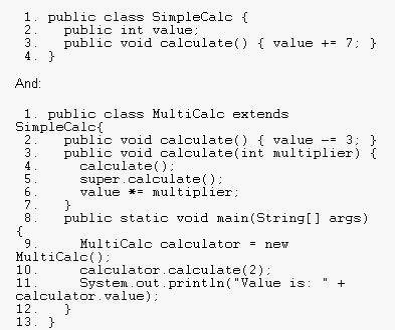
D. z.a1();

E. y.c1();

F. x.a1();

**Ques: 25 Click the Exhibit button.**

**What is the result?**



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.

**Ques: 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**