|  |
| --- |
| **Question 1 :**  Given:  11. public interface Status {  12. /\* insert code here \*/ **int MY\_VALUE = 10;**  13. }  Which three are valid on line 12? (Choose three.)  -> 인터페이스는 추상메서드와 상수를 가질 수 있다.  - >상수의 조건  1) 모든객체가 접근할 수 있어야 한다. public  2) 모든 인스턴스가 공유할 수 있어야 한다. static  3) 모든 상수를 .. final  **A. final**  **B. static**  C. native  **D. public**  E. private  F. abstract  G. protected |
| **Question 5 :**  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. }  -> 삼각형을 그려서 상속관계 파악해 보기  Which code, inserted at line 16, will cause a java.lang.ClassCastException?    A. Alpha a = x; // O  **B**. Foo f= (Delta)x; // X  C. Foo f= (Alpha)x; // O  D. Beta b = (Beta)(Alpha)x; //O |
| **Question 7 :**  Given:  20. public class CreditCard {  21.// 멤버변수 선언  22. private String cardlD;  23. private Integer limit;  24. public String ownerName; // 공개하면 안되는데 공개해버림!!  25.// 멤버메소드 초기화 선언  26. public void setCardlnformation(String cardlD,  27. String ownerName,  28. Integer limit) {  29. this.cardlD = cardlD;  30. this.ownerName = ownerName;  31. this.limit = limit;  32. }  33. }  Which is true? (올바른 것은?)  A. The class is fully encapsulated. // 완전히 은닉화 되었다. X  B. The code demonstrates polymorphism. // 다형성을 보여주고 있다. X  **C**. The ownerName variable breaks encapsulation. // 은닉화를 깨버렸다. O  D. The cardlD and limit variables break polymorphism. // 다형성 이야기를 해서 X  E. The setCardlnformation method breaks encapsulation. // 메소드가 은닉화를 깨버렸다. X |
| **Question 8 :**  Assume that country is set for each class.  // 은닉화문제) 직접적 접근 불가, 자식으로 인해 접근해야 됨.  Given:  10. public class Money {  11. private String country, name;  12. public String getCountry() { return country; }  13. }  and:  24. class Yen extends Money {  25. public String getCountry() { return super.country; } // 부모는 private이므로 직접적인 변수 접근 불가  26. }  27.  28. class Euro extends Money {  29. public String getCountry(String timeZone) {  30. return super.getCountry(); // 메소드를 통해서 간접적으로 접근한 후 부모가 자기자신을 호출하게 함. O  31. }  32. }    Which two are correct? (Choose two.)  A. Yen returns correct values. // 은닉화 위배 X  **B**. Euro returns correct values.// O  C. An exception is thrown at runtime. // 실행타임이 아닌 실행하기도 전에 이미 문법검사에서 에러발생 X,  // runtime이 아닌 compilation  D. Yen and Euro both return correct values. // X  **E**. Compilation fails because of an error at line 25. // O  F. Compilation fails because of an error at line 30. // X |
| **Question 9 :**  Which Man class properly represents the relationship “Man has a best friend who is a Dog”?  **-> 개가 “자료형” 이다.**  A. class Man extends Dog { }  B. class Man implements Dog { } // 개는 사물이 아니기 때문에 추상적은 거리가 멀다. 그러나 개의 행동  // 같은 상태는 가능하다.  C. class Man { private BestFriend dog; }  **D**. class Man { private Dog bestFriend; } // has a 관계  E. class Man { private Dog<bestFriend> }  F. class Man { private BestFriend<dog> } |
| **Question 12 :**  12. Given:  // CALL by values  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. doStuffx = 6 main x = 6  **D.** doStuffx = 5 main x = 5 // O  E. doStuffx = 5 main x = 6  F. doStuffx = 6 main x = 5 |
| **Question 15 :**  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; } } |
| **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 { } |
| **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 20 :**  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. |
| **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();  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(); |
| **Question 22 :**  Given:  10. abstract public class Employee {  11. protected abstract double getSalesAmount();  12. public double getCommision() {  13. return getSalesAmount() \* 0.15;  14. }  15. }  16. class Sales extends Employee {  17. // insert method here  18. }  Which two methods, inserted independently at line 17, correctly complete the Sales class? (Choose two.)  A. double getSalesAmount() { return 1230.45; }  B. public double getSalesAmount() { return 1230.45; }  C. private double getSalesAmount() { return 1230.45; }  D. protected double getSalesAmount() { return 1230.45; } |
| **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\*/ }  }  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() { /\*d 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\*/ } |
| **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 29 :**  Given:  55. int []x= {1,2,3,4,5};  56. int y[] = x;  57. System.out.println(y[2]);  Which is true?  A. Line 57 will print the value 2.  B. Line 57 will print the value 3.  C. Compilation will fail because of an error in line 55.  D. Compilation will fail because of an error in line 56. |
| **Question 30 :**  Given:  35. String #name = “Jane Doe”;  36. int $age=24;  37. double \_height = 123.5;  38. double ~temp = 37.5;  Which two are true? (Choose two.)  A. Line 35 will not compile.  B. Line 36 will not compile.  C. Line 37 will not compile.  D. Line 38 will not compile. |
| **Question 35 :**  Given:  11. public static void main(String[] args) {  12. Object obj =new int[] { 1,2,3 };  13. int[] someArray = (int[])obj;  14. for (int i: someArray) System.out.print(i +“ “)  15. }  What is the result?  A. 1 2 3  B. Compilation fails because of an error in line 12.  C. Compilation fails because of an error in line 13.  D. Compilation fails because of an error in line 14.  E. A ClassCastException is thrown at runtime. |
| **Question 36 :**  Given:  10. class Foo {  11. static void alpha() { /\* more code here \*/ }  12. void beta() { /\* more code here \*/ }  13. }  Which two are true? (Choose two.)  A. Foo.beta() is a valid invocation of beta().  B. Foo.alpha() is a valid invocation of alpha().  C. Method beta() can directly call method alpha().  D. Method alpha() can directly call method beta(). |
| **Question 42 :**  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 45 :**  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. |
| **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.  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. |
|  |