Practice OCA :

1. Mảng là các đối tượng vì vậy tham chiếu được thay đổi trong method gọi đến mảng đó được phản ảnh mọi nơi mà mảng được sử dụng.
2. Biến nguyên thủy được truyền giá trị , trong phương thức khi gọi tham số đã truyền một biến nguyên thủy khác với biến trong phương thức vì vậy thành viên thể hiện không thay đổi giá trị
3. Java.lang.Exception là một checked Exception có nghĩa là , phương thức throw ngoại lệ này phải khai báo nó trong mệnh đề throws do đó phương thức của bạn phải khai báo throws exception.
4. Object[] oa = new Object[10] , oa chứa 10 giá trị mặc định của object khai báo
5. Overload phương thức cùng tên khác tham số hoặc thứ tự , số tham số , kiểu trả về không quan trọng
6. RuntimeException ko cần phải handle trong khi Exception phải handle trong try catch or throws
7. Class chỉ khai bảo public khi trùng tên file.
8. Final class ko thể kế thừa , nhưng đc kế thừa class khác , phương thức final ko thể override nhưng có thể kế thừa , biến final ko thể thay đổi giá trị , nếu biến final trống đc khởi tạo bằng constructor và static final trống khởi tạo bằng static block
9. Static block code chạy trước rồi đến instance block rồi đến constructor , trong trg hợp kế thừa thì super luôn chạy trc nhưng ưu tiên nhất là static block code -> main -> super -> sub-class
10. Phương thức static có thể dùng chung quản lý bởi class ko phải instance , biến static the same.
11. Từ khóa native chỉ dùng cho phương thức
12. Quy tắc Switch : Chỉ String , byte , char , short , int và các wrapper Class của nó , có cả enums được sử dụng làm kiểu một biến chuyển đổi , Chuỗi đc cho phép từ Java 7 . Các hằng số case phải được gán biến switch, nếu biến switch là String , biến case là String . Biến Switch phải đủ lớn để chứa tất cả các hằng số case , ví dụ biến char thì không có hằng số case nào vượt quá 65535 vì scope của char là 0 – 65535. Tất cả các biến case phải là constants. Không có hai trong số các biểu thức hằng số case liên quan đến biến switch có thể cùng giá trị. Nhiều nhất một biến default có thể liên kết với biến switch.
13. Mảng là một đối tượng , nhưng mảng của đối tượng không thể là mảng của các đối tượng khác. Object[] obj = new int[10] là sai. Size của mảng phải được khai báo bên phải. Int [] i ,j là I , j đều là mảng int , int I [] , j là I là mảng int còn j là số int.
14. Mảng đa chiều , size đầu tiên phải được chỉ định , các size sau có thể không cần chỉ định vì mảng đa chiều ko cần phải đối xứng. Mảng một chiều độ dài phải được chỉ định.
15. Các biến chứa tham chiếu đến các đối tượng thực tế . Việc chỉ định cho một tham chiếu đến đối tượng thực tế khác chỉ thay đổi giá trị của tham chiếu và không thay đổi giá trị đối tượng thực tế , toString override tại lớp String chỉ in ra chuỗi thực tế.
16. Bắt đầu từ java 7 bạn có thể sử dụng dấu gạch dưới giữa các số để giúp viết số dài dễ hơn ví dụ nếu viết 1000000 có thể viết 1\_000\_000. Lưu ý bạn không thể bắt đầu hoặc kết thúc một giá trị với dấu gạch dưới do đó 100\_ và \_100 là các giá trị không hợp lệ, \_100 là tên biến hợp lệ , bạn có thể sử dụng dấu gạch dưới cho tất cả các loại số bao gồm long , double , binary , float , hex ví dụ : int hex = 0xCAFE\_BASE , float f = 9898\_7878.333\_333f , int bin = 0b111\_0000\_1100\_1100;
17. Biến nào (hoặc phương thức tĩnh) sẽ được sử dụng tùy thuộc vào lớp mà biến được khai báo. Phương thức cá thể nào sẽ được sử dụng phụ thuộc vào lớp thực tế của đối tượng được tham chiếu bởi biến.
18. String trong cùng một lớp cùng 1 package hoặc trong các lớp khác nhau cùng 1 package hoặc các lớp khác nhau trong các package khác nhau biểu thị các tham chiếu đến cùng một đối tượng String , Các chuỗi được tính bằng các biểu thức không đổi được tính toán tại thời điểm biên dịch và sau đó được xử lý nếu chúng là chữ. Chuỗi được tính toán trong thời gian chạy run time , được tạo mới và do đó nó là khác biệt, Kết quả của việc thực hiện intern() một chuỗi là cùng một chuỗi bất kỳ chuỗi kí tự có sẵn nào có cùng nội dung.
19. Thực hiện toán tử luôn thực hiện bên trái trước rồi đến bên phải.
20. String , StringBuilder , StringBuffer đều là các final class , Boolean , Integer , Long , Short là final , Number không phải là final , int , long , double extended Number , System là final
21. Có các trường hoặc phương thức mơ hồ không tự gây ra bất kỳ vấn đề nào nhưng việc tham khảo các trường / phương thức đó theo cách không rõ ràng sẽ gây ra lỗi thời gian biên dịch. Vì vậy, bạn không thể gọi: System.out.println (VALUE); bởi vì nó sẽ mơ hồ (có hai định nghĩa GIÁ TRỊ). Nhưng các dòng sau là hợp lệ: TestClass tc = new TestClass (); System.out.println (((T1) tc) .VALUE); Tuy nhiên, không cần phải sử dụng cast rõ ràng để gọi phương thức m1 (): ((T2) tc) .m1 (); tc.m1 () cũng tốt vì mặc dù m1 () được khai báo trong cả hai giao diện, định nghĩa cho cả hai đều giải quyết rõ ràng chỉ một m1 (), được định nghĩa trong TestClass.
22. Để ghi đè một phương thức trong lớp con, phương thức ghi đè (nghĩa là phương thức trong lớp con) PHẢI CÓ: cùng một kiểu trả về trong trường hợp nguyên thủy (một lớp con được phép cho các lớp, đây còn được gọi là kiểu trả về covariant). cùng loại và thứ tự của các tham số. nó chỉ có thể ném ra những ngoại lệ được khai báo trong mệnh đề ném của phương thức của siêu lớp hoặc các ngoại lệ là các lớp con của các ngoại lệ được khai báo. Nó cũng có thể chọn KHÔNG ném bất kỳ ngoại lệ. Tên của các loại tham số không quan trọng. Ví dụ, void methodX (int i) giống như void methodX (int k)