

Lab exercise

1. Can one access instance attributes/methods in a static method?
2. Can one access static attributes/methods in a instance method?
3. Can one access static member of any other class in a static method?
4. Can an instance method override static method?
5. Can an instance attribute hide static attribute?
6. When should an abstract Class Implement an Interface?
7. Can an abstract class have static attributes and static methods?
8. When does one choose an abstract class over an interface?
9. Is following code valid?

```
class A
{
    int x = 0;
    int getX(){return x};
}
class B extends A
{
    float x = 0.0f;
    float getX(){return x};
}
```

Case Study

Given the following write the java code for the following, compile and run it.

Step 1:- There is a class Vehicle. It has a public method start().

write down the two subclasses for this vehicle in same package as

a:- Class Car, with public method driving()

b:- Class Aircraft, with public method flying().

Provide some messages in all the methods so that when they are called you get the desired output printed.

Now, write down one more class say MyApplication to instantiate these classes. Compile and run it using Netbeans.

Step 2:- Now try the multilevel Inheritance by creating subclasses for Aircraft class in same package as

a:- Class Jet, with public method zoom().

b:- Class Whirlbird, with public method whirl().

Provide some messages in all the methods so that when they are called you get the desired output printed.

Now, try to instantiate these new classes by changing the class MyApplication. Compile and run it.

Step 3:- Check, what happens if we change the access modifier of start() in class Vehicle from public to private.

First think, and explain the reason, then try it out by Compiling and running it.

Step 4:- Check, what happens if we change the access modifier of method start() in class Vehicle to protected.

Think and explain the reason. Try it out by Compiling and running it using Netbeans.

Step 5:- Now try to override the start() of Vehicle Class in any of the subclass, say Class Car.

Print the output of Car class start() method.

What changes you have to do, to call the start() of Vehicle class from Car class?

First think then try, explain the reason.

Quiz

1. In object oriented programming a class of objects can _____ properties from another class of objects.
2. When the action referred to in question one does occur, the new class is called a _____ .
3. When the action referred to in question one occurs, the originating class is called a _____ .
4. By hiding the implementation details of an object the only thing that remains externally visible is the _____ of the object.
5. The term given to the process of hiding all the details of an object that do not contribute to its essential characteristics is called _____.
6. What is an Object?
7. What is a Class?
8. What is an Abstract Class?
9. What is a SubClass?
10. What is Inheritance?
11. Difference between Overloading and Overriding?
12. What is a final Class?
13. What is Interface?
14. What is the difference b/w abstract class & interface?
15. What does Java Language allow multiple inheritance.?
16. What are packages and what are they used for?
17. How do Objects pass messages in Java?
 - a:- by modifying each others filed
 - b:- by modifying the static variables of each other's classes.
 - c:- by calling each other's instance methods.
 - d:- by calling static methods of each other's classes.
18. _____ is a feature of OOP.
19. Which is characterized by the phrase 'one interface, multiple methods'.
20. When you declare the object of a class, you are creating _____ of the class.

21. Consider the following three cases:
- a:- Class A is associated with class B
 - b:- Class C depends on Class D
 - c:- Class E is a composite of classes
22. Write a marginal Java code related to them?
23. All methods in an abstract class must be declared abstract (true/false)
- 24.
25. Subclasses of an abstract class that do not provide an implementation of an abstract method , are also abstract (true/false).
-