

1. A superclass knows nothing about its subclass's fields

- A True
- B False

2. `int i;`  
`Vehicle v = new Car();`

The static type of `v` is:

- A Vehicle
- B Car
- C int
- D null
- E notype

3. `int i;`  
`Vehicle v = new Car();`

The dynamic type of `v` is:

- A Vehicle
- B Car
- C int
- D null
- E notype

4. For overriding to take place, the superclass and subclass must both define methods - with the same signature

- A True
- B False

5. An overridden method can't be called from the method that overrides it

- A True
- B False

```
6. public void display()
{
super.display();
System.out.println(" [" + message + "]");
}
```

**What does the code above do?**

- A It recursively calls the display method
- B It calls display() twice
- C it calls the display method in the superclass (which can access the fields of that class and print them out), then prints out the message field from the current class.
- D it ignores the display method in the superclass and prints out the message field from the current class.

**7. super() ...**

- A is the constructor for the Super class
- B is called by Java to let you know everything is good i.e. super!
- C calls the parent class constructor
- D calls the superclass class constructor
- E calls the subclass class constructor

**8. instanceof is used to**

- A create an instance variable
- B determine the static type of a variable
- C determine the dynamic type of a variable

**9. All classes inherit from**

- A Application
- B Class
- C Object
- D Origin
- E Super

**10. the protected access modifier enables**

- A full access by all
- B Subclass methods to access the fields of the class they inherit from
- C all methods to access the fields of the class
- D methods to access the fields of the class they are in
- E all methods in all classes have access