Core Java test1

1. An object can have any number of references. Yes/no
2. We can create any number of objects to one class. Yes/no
3. Which access comes in between default and public?
4. One reference can refer any number of objects in the same statement. Yes/no
5. Constructor can be static. Yes/no
6. Which are the calling statements allowed only in the constructors?
7. Constructor should have a return type. Yes/no
8. Constructor has any name. yes/no
9. First statement of the constructor is any kind of this() or any kind of super(). Yes/no
10. Methods should not have a return type. Yes/no
11. Which access members cant be used in the outside a subclass while its inheriting from another package member?
12. What are the return values if method return type is boolean?
13. What should be the return type if method should not return a value?
14. Which access level is only for the class level?
15. Which access is for within the package?
16. Which access is the widermost?
17. Which concept is used to avoid naming collisions?
18. Which is the keyword acting as a reference variable?
19. Constructor inherits to the subclass. Yes/no?
20. IIB inherits to the subclass. Yes/no?
21. SIB inherits to the subclass. Yes/no?
22. attributes inherits to the subclass. Yes/no?
23. methods inherits to the subclass. Yes/no?
24. abstract class should have min one abstract method. Yes/no
25. we can create an object to an abstract class. Yes/no
26. we can create an object reference to an abstract class. Yes/no
27. we cant develop constructor inside an abstract class. Yes/No
28. constructor can be an abstract. Yes/no
29. Which is the access level to the members of an interface?
30. Which access level keywords are not allowed to the interface members?

Ans:

1. Yes
2. Yes
3. protected
4. No
5. No
6. this() (or) super()
7. No
8. No
9. Yes
10. No
11. protected
12. true (or) false
13. void
14. private
15. default scope
16. public
17. packages
18. this
19. no
20. no
21. no
22. yes
23. yes
24. no
25. no
26. yes
27. no
28. no
29. public
30. private and protected