**2) Can you create a sub class to the following class?**

class A

{

private A()

{

//First Constructor

}

private A(int i)

{

//Second Constructor

}

}

# No, you can’t create sub classes to that class which has only private constructors.

**3) Can you find out the error in the below code?**

private class A

{

private class B

{

//Inner class

}

}

# Inner class can be private. But outer class cannot be private

**4) Does method ‘iMethod’ of Class A be inherited to Class B in the below code?**

class A

{

protected void iMethod(){

};

}

class B extends A

{

}

#Yes can be inherited as it is protected

**5) Is the below code written correctly?**

class A

{

private class B

{

//inner class

}

}

public class MainClass extends A

{

public static void main(String[] args)

{

B b = new B();

}

}

# No. private inner Class B can not be instantiated outside the Class A.

**6) Is the below code written correctly?**

package pack1;

class A

{

}

package pack2;

class B extends A

{

}

# class is having modifier default hence cannot be inherited outside pack1

**7) Can we declare a class as protected?**

#Yes but only inner class outer class cannot be protected

**8) Do you think the below program is written correctly? If yes, what will be the output?**

package pack1;

class X

{

protected int i = 1221;

void methodOfX()

{

System.out.println(i);

}

}

public class MainClass

{

public static void main(String[] args)

{

X x = new X();

System.out.println(x.i);

x.methodOfX();

}

}

1221  
1221

**9) Why we can’t instantiate Class-A in the below code outside the package even though it has public constructor?**

package pack1;

class A

{

public A()

{

//public constructor

}

}

package pack2;

import pack1.\*;

class B

{

A a = new A(); //Compile Time Error

}

# Class A is having modifier as default or no modifier

**10) Can a protected field of a class be inherited to subclass outside the package?**

**11) Why the below code is throwing compile time error?**

package pack1;

public class A

{

protected A()

{

//protected constructor

}

}

package pack2;

import pack1.A;

class B

{

A a = new A(); //Compile time error

}

# Constructor is protected hence cannot be called outside package

**12) Do you think the below code compiles successfully even though it is calling super class’s protected constructor outside the package?**

package pack1;

public class A

{

protected A(int i)

{

//protected constructor

}

}

package pack2;

import pack1.A;

class B extends A

{

public B()

{

super(10); //calling super class's protected constructor

}

}

# Yes it will work fine

**13) Can we declare static methods as private?**

#Yes can be

**14) Is the below code written correctly? If yes, what will be the output?**

package pack1;

class A

{

protected static String s = "A";

}

class B extends A

{

}

class C extends B

{

static void methodOfC()

{

System.out.println(s);

}

}

public class MainClass

{

public static void main(String[] args)

{

C.methodOfC();

}

}

#Yes, it is written correctly. Output will be A.

**15) Write the access modifiers in the increasing order of their visibility?**

private —> default or no access modifiers —> protected —> public

**16) How many public classes a .java file can have?**

Only one

**17) What will be the outcome of the below program?**

package pack1;

public class A

{

private int methodOne(int i)

{

return ++i;

}

public int methodTwo(int i)

{

return methodOne(++i);

}

}

package pack2;

import pack1.A;

class B extends A

{

int methodOne(int i)

{

return methodTwo(++i);

}

}

public class MainClass

{

public static void main(String[] args)

{

System.out.println(new B().methodOne(101));

}

}

104

**18) Can you find out the error in the following code snippet?**

class A

{

public void methodOfA()

{

System.out.println("Class A");

}

}

class B extends A

{

@Override

void methodOfA()

{

System.out.println("Class B");

}

}

# The visibility of methodOfA() has been reduced to default while overriding it in the class B. You can’t reduce the visibility of a method while overriding it.

**19) private method can be overridden as public method. True or False?**

False. private methods are not at all inherited.

**20) A method of super class with default access modifier can be overridden as protected or public but not as private. True or false?**

**#**True

**21) Monu has written the code like below but it is showing compile time error. Can you help him to remove the error?**

private class A

{

private class B

{

private class C

{

}

}

}

#Outer class can’t be private. Don’t declare Class A as private