```
-------------------------------------
//  the clone() method is protected in java.lang.Object. So, the clone() method cannot
//  be called on an object that does not implement it

public class CloneMe {
    private int identifier = 0;

    public CloneMe(int identifierIn) {
        identifier = identifierIn;
    }


    // public Object clone() {
    //  return new CloneMe(17);
    // }

    public String toString() {
        String returnString = "\n Clone Me " + identifier;
        return returnString;
    }
}



public class CloneTest {
    public CloneTest() {}

    public void doTest() {
        CloneMe c = new CloneMe(19);
        CloneMe anotherC = (CloneMe)c.clone();
        System.out.println(c);
        System.out.println(anotherC);
    }

    public String toString() {
        String returnString = "\n Clone Test";
        return returnString;
    }

}



public class Driver {

    public static void main(String[] args) {

        CloneTest cTest = new CloneTest();
        cTest.doTest();
    }

}



bravo:~/testJavaClone>  javac *.java
CloneTest.java:7: clone() has protected access in java.lang.Object
        CloneMe anotherC = (CloneMe)c.clone();
                                    ^
1 error
bravo:~/testJavaClone>
```

```
 ---------------------------------------------------------------------

 //  the clone() method is protected in java.lang.Object. So, the clone() method CAN
 //  be called on an object that overrides the method.



 public class CloneTest {
     public CloneTest() {}

     public void doTest() {
         CloneMe c = new CloneMe(19);
         CloneMe anotherC = (CloneMe)c.clone();
         System.out.println(c);
         System.out.println(anotherC);
     }

     public String toString() {
         String returnString = "\n Clone Test";
         return returnString;
     }

 }
 public class CloneMe {
     private int identifier = 0;

     public CloneMe(int identifierIn) {
         identifier = identifierIn;
     }


     public Object clone() {
         return new CloneMe(17);
     }

     public String toString() {
         String returnString = "\n Clone Me " + identifier;
         return returnString;
     }
 }


 public class Driver {

     public static void main(String[] args) {

         CloneTest cTest = new CloneTest();
         cTest.doTest();
     }

 }


 bravo:~/testJavaClone> javac *.java

 bravo:~/testJavaClone> java Driver

  Clone Me 19
```

```
 Clone Me 17

bravo:~/testJavaClone>

---------------------------------------------------------------------

// If the Cloneable interface is implemented, it is possible to test if a reference
// is marked/tagged as an instance of Cloneable


public class CloneMe implements Cloneable{
    private int identifier = 0;

    public CloneMe(int identifierIn) {
        identifier = identifierIn;
    }


    public Object clone() {
        return new CloneMe(17);
    }

    public String toString() {
        String returnString = "\n Clone Me " + identifier;
        return returnString;
    }
}


public class CloneTest {
    public CloneTest() {}

    public void doTest() {
        CloneMe c = new CloneMe(19);
        System.out.println(c);

        if (c instanceof Cloneable) {
            CloneMe anotherC = (CloneMe)c.clone();
            System.out.println(anotherC);
        }
    }

    public String toString() {
        String returnString = "\n Clone Test";
        return returnString;
    }

}

public class Driver {

    public static void main(String[] args) {

        CloneTest cTest = new CloneTest();
        cTest.doTest();
    }

}



bravo:~/testJavaClone> javac *.java
bravo:~/testJavaClone> java Driver
```

```
 Clone Me 19

 Clone Me 17
bravo:~/test/testJava/clone>


------------------------------------------------------------------------
// If you mark/tag a class has Cloneable it
// is NOT required to override the clone method


public class CloneMe implements Cloneable{
    private int identifier = 0;

    public CloneMe(int identifierIn) {
        identifier = identifierIn;
    }


    // public Object clone() {
    //  return new CloneMe(17);
    // }

    public String toString() {
        String returnString = "\n Clone Me " + identifier;
        return returnString;
    }
}




public class CloneTest {
    public CloneTest() {}

    public void doTest() {
        CloneMe c = new CloneMe(19);
        System.out.println(c);

        if (c instanceof Cloneable) {
            //      CloneMe anotherC = (CloneMe)c.clone();
            //      System.out.println(anotherC);
        }
    }

    public String toString() {
        String returnString = "\n Clone Test";
        return returnString;
    }

}


public class Driver {

    public static void main(String[] args) {

        CloneTest cTest = new CloneTest();
        cTest.doTest();
    }

}
```

```
bravo:~/testJavaClone> javac *.java
bravo:~/testJavaClone> java Driver

 Clone Me 19
bravo:~/test/testJava/clone>
```