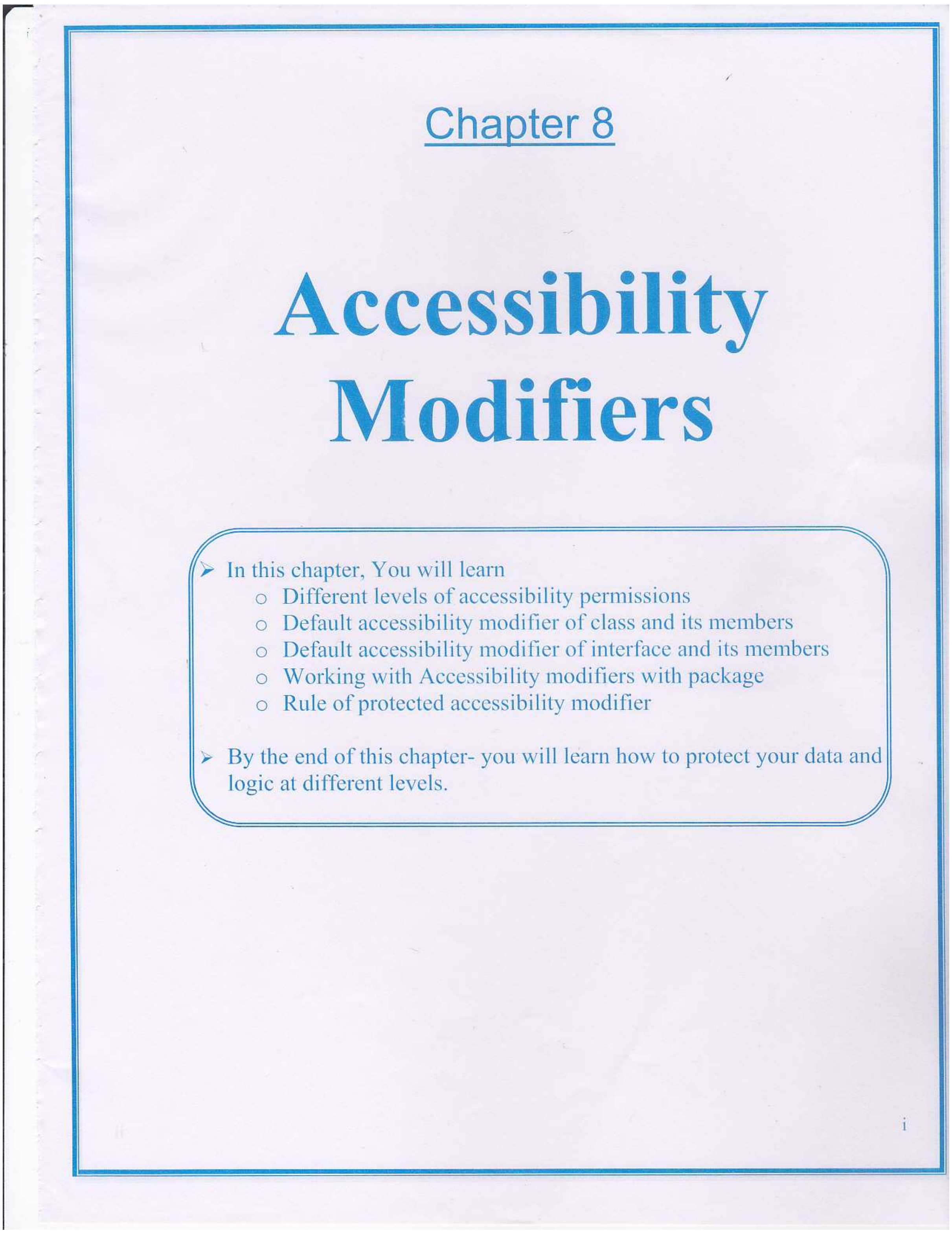
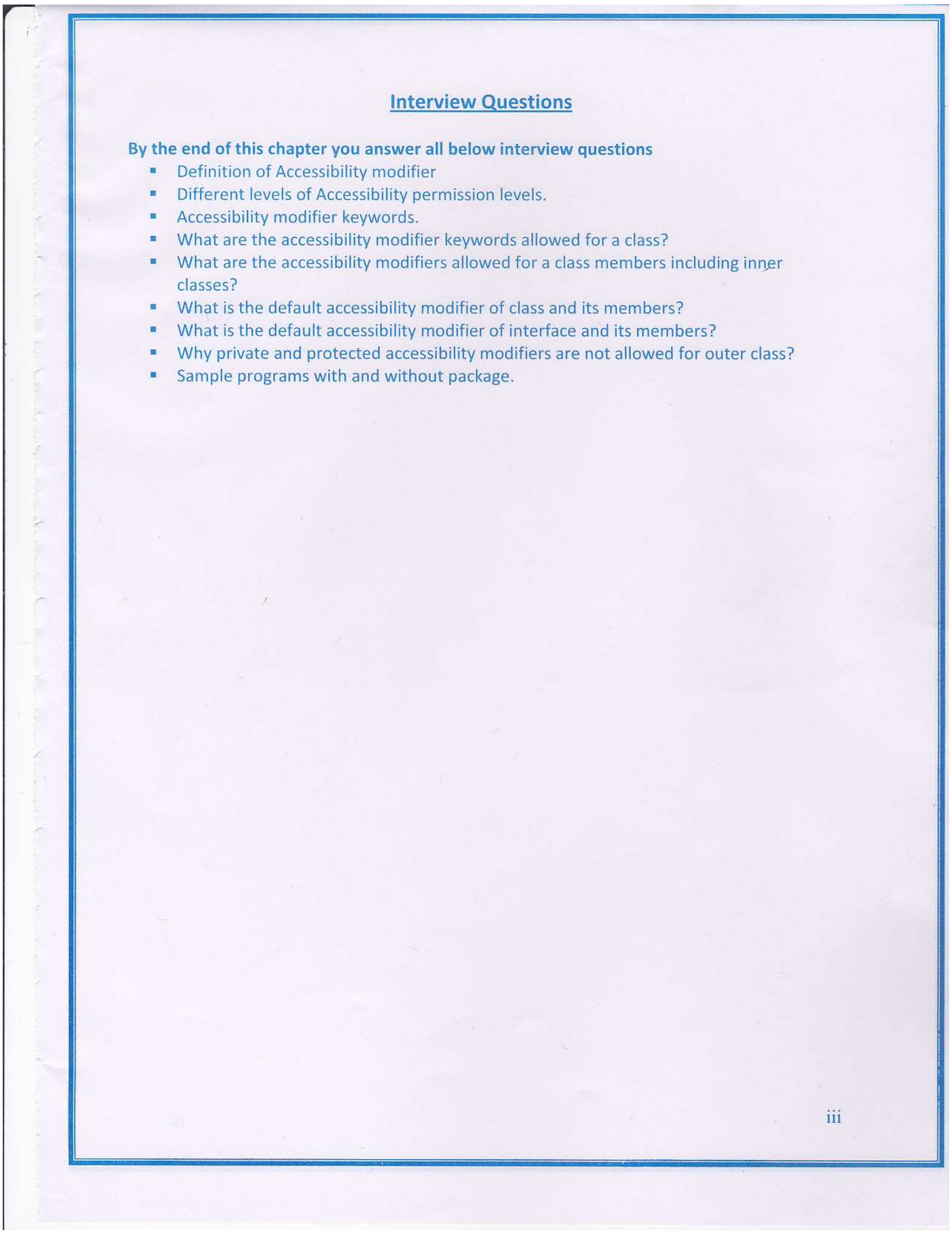


Chapter 8

# Accessibility Modifiers

- In this chapter, You will learn
  - Different levels of accessibility permissions
  - Default accessibility modifier of class and its members
  - Default accessibility modifier of interface and its members
  - Working with Accessibility modifiers with package
  - Rule of protected accessibility modifier
- By the end of this chapter- you will learn how to protect your data and logic at different levels.

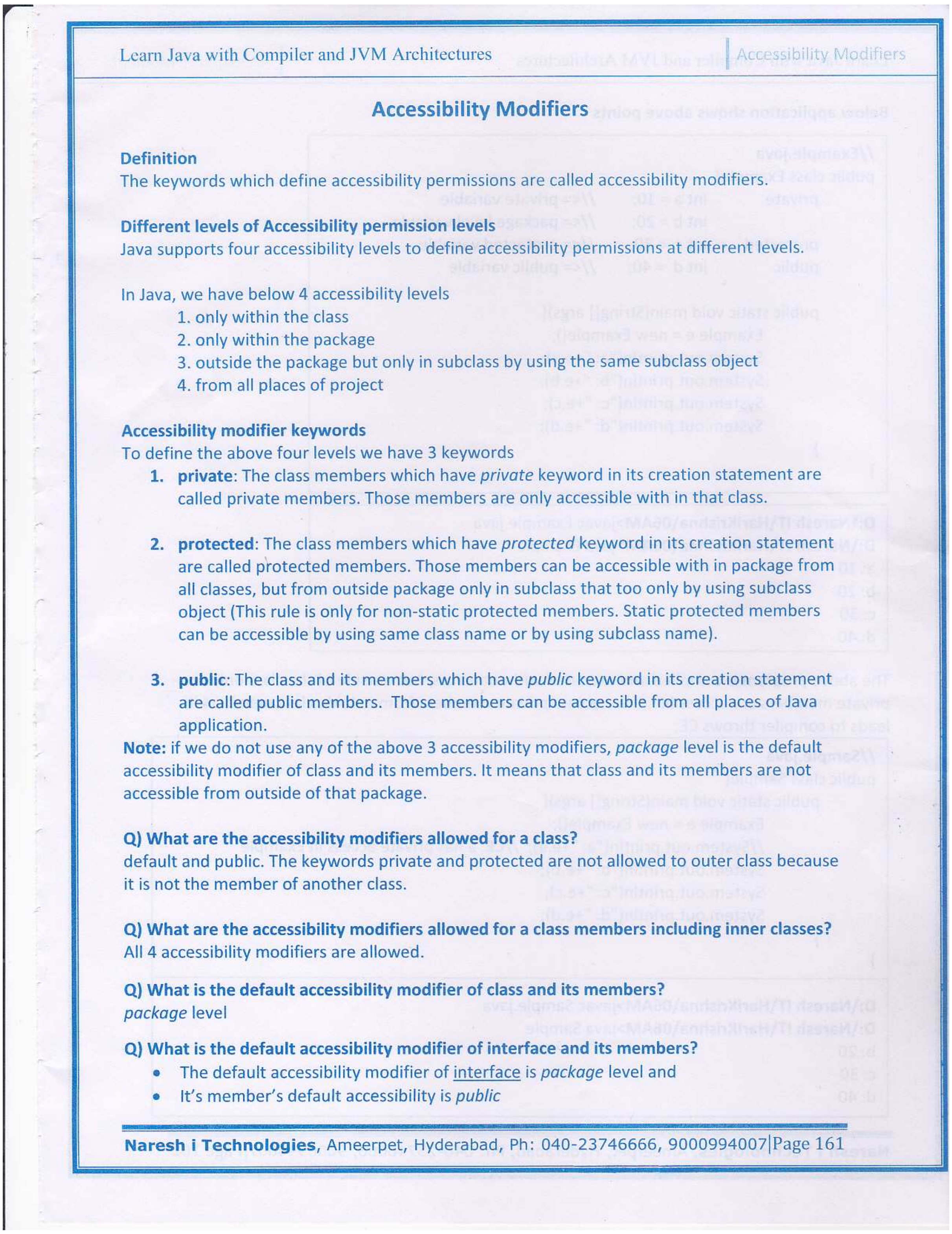


## Interview Questions

**By the end of this chapter you answer all below interview questions**

- Definition of Accessibility modifier
- Different levels of Accessibility permission levels.
- Accessibility modifier keywords.
- What are the accessibility modifier keywords allowed for a class?
- What are the accessibility modifiers allowed for a class members including inner classes?
- What is the default accessibility modifier of class and its members?
- What is the default accessibility modifier of interface and its members?
- Why private and protected accessibility modifiers are not allowed for outer class?
- Sample programs with and without package.

iii



## Accessibility Modifiers

**Definition**
The keywords which define accessibility permissions are called accessibility modifiers.

**Different levels of Accessibility permission levels**
Java supports four accessibility levels to define accessibility permissions at different levels.

In Java, we have below 4 accessibility levels
1. only within the class
2. only within the package
3. outside the package but only in subclass by using the same subclass object
4. from all places of project

**Accessibility modifier keywords**
To define the above four levels we have 3 keywords

1. **private**: The class members which have *private* keyword in its creation statement are called private members. Those members are only accessible with in that class.

2. **protected**: The class members which have *protected* keyword in its creation statement are called protected members. Those members can be accessible with in package from all classes, but from outside package only in subclass that too only by using subclass object (This rule is only for non-static protected members. Static protected members can be accessible by using same class name or by using subclass name).

3. **public**: The class and its members which have *public* keyword in its creation statement are called public members. Those members can be accessible from all places of Java application.

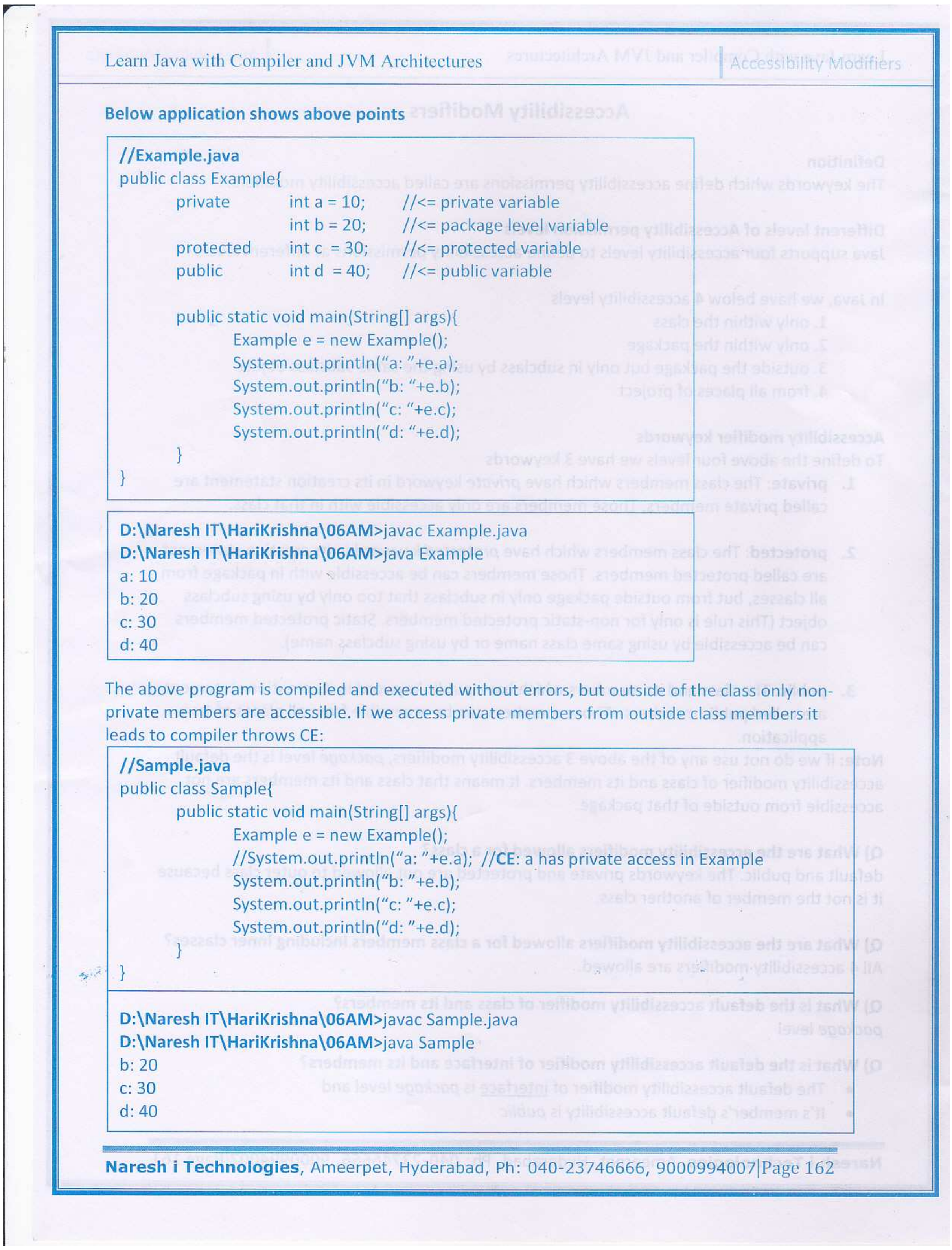
**Note:** if we do not use any of the above 3 accessibility modifiers, *package* level is the default accessibility modifier of class and its members. It means that class and its members are not accessible from outside of that package.

**Q) What are the accessibility modifiers allowed for a class?**
default and public. The keywords private and protected are not allowed to outer class because it is not the member of another class.

**Q) What are the accessibility modifiers allowed for a class members including inner classes?**
All 4 accessibility modifiers are allowed.

**Q) What is the default accessibility modifier of class and its members?**
*package* level

**Q) What is the default accessibility modifier of interface and its members?**
- The default accessibility modifier of interface is *package* level and
- It's member's default accessibility is *public*



**Below application shows above points**

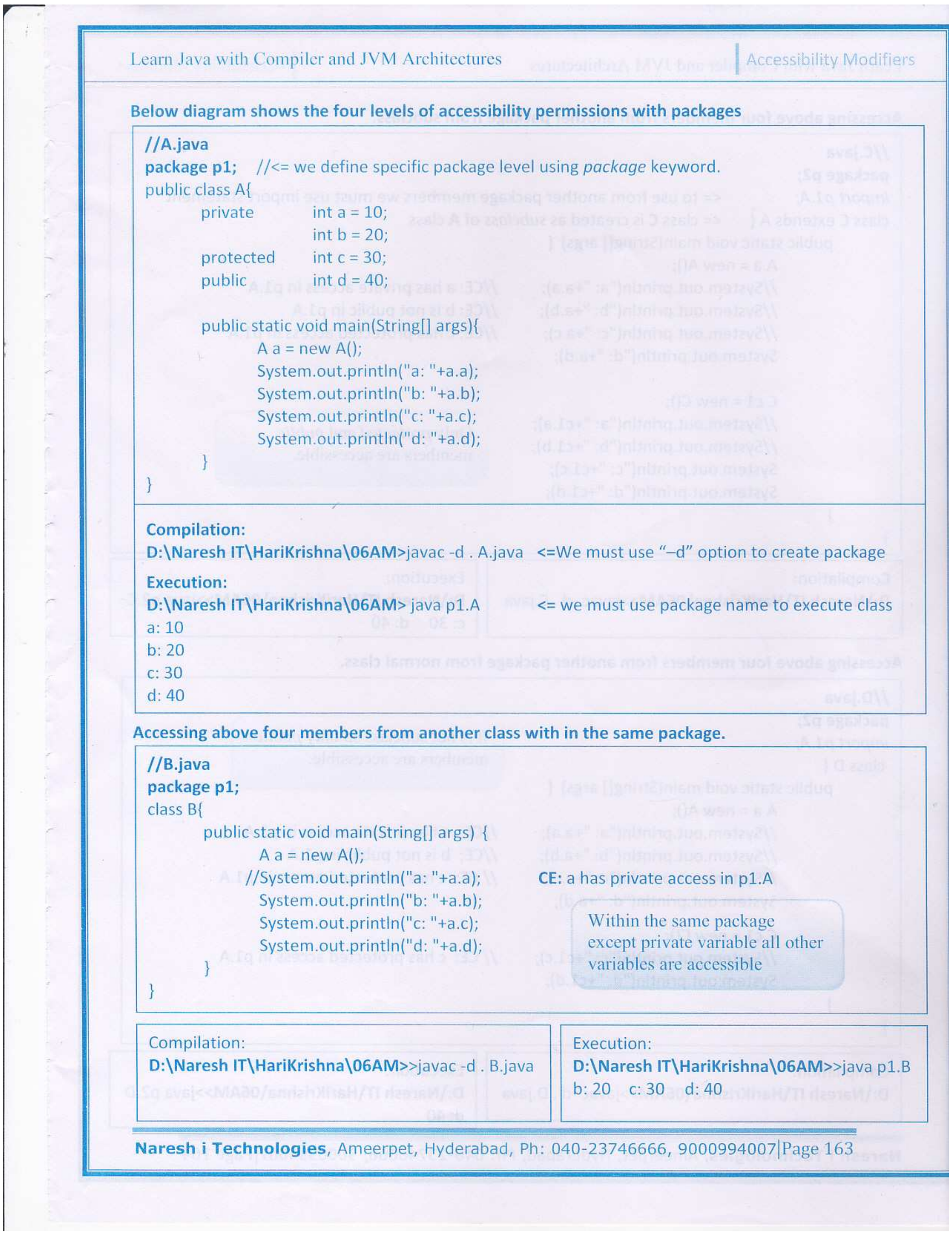
```
//Example.java
public class Example{
          private         int a = 10;       //<= private variable
                          int b = 20;       //<= package level variable
          protected       int c = 30;       //<= protected variable
          public          int d  = 40;      //<= public variable

          public static void main(String[] args){
                    Example e = new Example();
                    System.out.println(“a: ”+e.a);
                    System.out.println(“b: “+e.b);
                    System.out.println(“c: “+e.c);
                    System.out.println(“d: “+e.d);
          }
}
```

```
D:\Naresh IT\HariKrishna\06AM>javac Example.java
D:\Naresh IT\HariKrishna\06AM>java Example
a: 10
b: 20
c: 30
d: 40
```

The above program is compiled and executed without errors, but outside of the class only non-private members are accessible. If we access private members from outside class members it leads to compiler throws CE:

```
//Sample.java
public class Sample{
          public static void main(String[] args){
                    Example e = new Example();
                    //System.out.println(“a: ”+e.a);  //CE: a has private access in Example
                    System.out.println(“b: ”+e.b);
                    System.out.println(“c: ”+e.c);
                    System.out.println(“d: ”+e.d);
          }
}
```

```
D:\Naresh IT\HariKrishna\06AM>javac Sample.java
D:\Naresh IT\HariKrishna\06AM>java Sample
b: 20
c: 30
d: 40
```



**Below diagram shows the four levels of accessibility permissions with packages**

```
//A.java
package p1;    //<= we define specific package level using package keyword.
public class A{
          private        int a = 10;
                         int b = 20;
          protected      int c = 30;
          public         int d = 40;

          public static void main(String[] args){
                    A a = new A();
                    System.out.println("a: "+a.a);
                    System.out.println("b: "+a.b);
                    System.out.println("c: "+a.c);
                    System.out.println("d: "+a.d);
          }
}
```

**Compilation:**

```
D:\Naresh IT\HariKrishna\06AM>javac -d . A.java   <=We must use "–d" option to create package
```

**Execution:**

```
D:\Naresh IT\HariKrishna\06AM> java p1.A          <= we must use package name to execute class
a: 10
b: 20
c: 30
d: 40
```

**Accessing above four members from another class with in the same package.**

```
//B.java
package p1;
class B{
          public static void main(String[] args) {
                    A a = new A();
                  //System.out.println("a: "+a.a);     CE: a has private access in p1.A
                    System.out.println("b: "+a.b);
                    System.out.println("c: "+a.c);
                    System.out.println("d: "+a.d);
          }
}
```

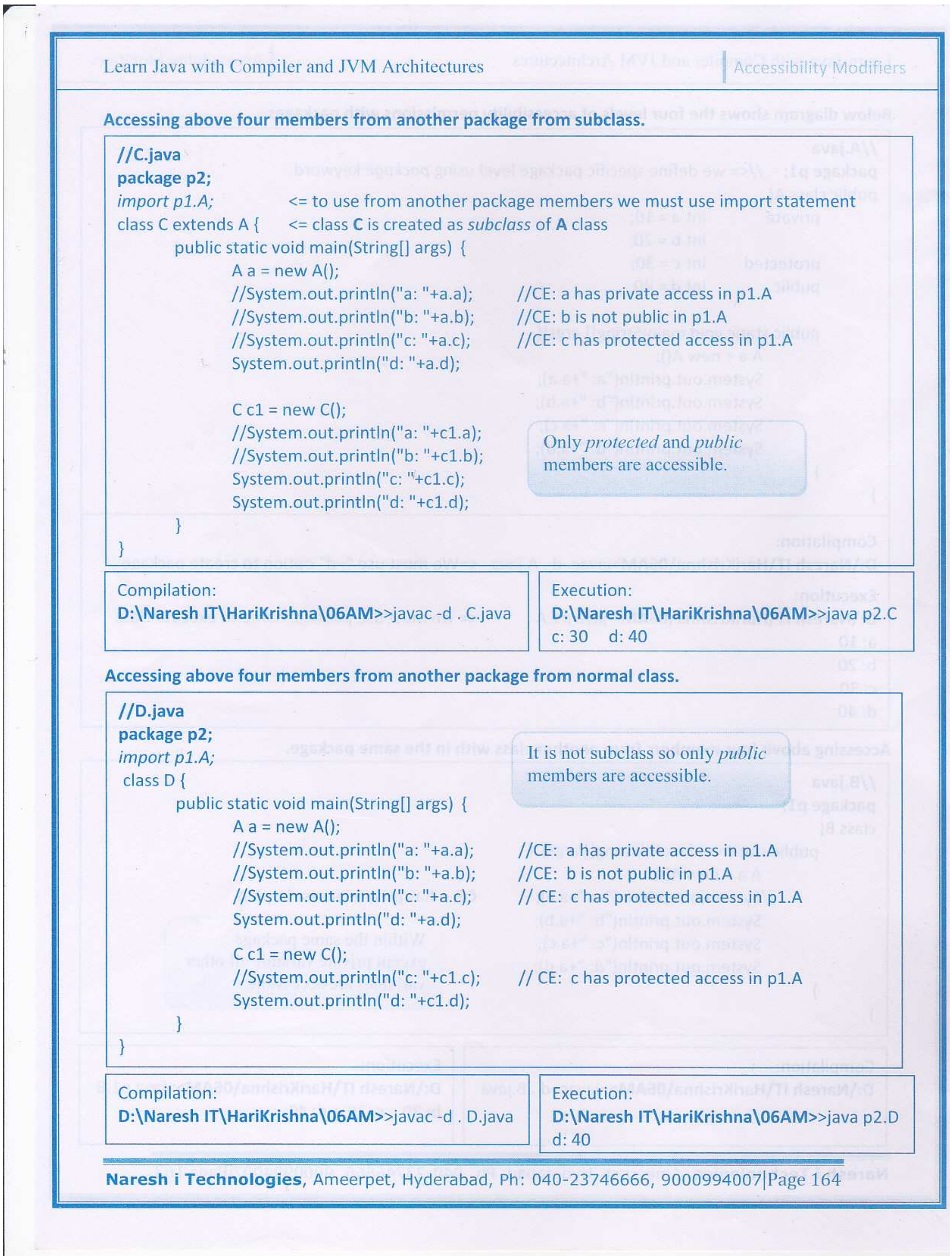
Within the same package except private variable all other variables are accessible

Compilation:

```
D:\Naresh IT\HariKrishna\06AM>>javac -d . B.java
```

Execution:

```
D:\Naresh IT\HariKrishna\06AM>>java p1.B
b: 20    c: 30    d: 40
```



**Accessing above four members from another package from subclass.**

```
//C.java
package p2;
import p1.A;              <= to use from another package members we must use import statement
class C extends A {       <= class C is created as subclass of A class
        public static void main(String[] args) {
                A a = new A();
                //System.out.println("a: "+a.a);     //CE: a has private access in p1.A
                //System.out.println("b: "+a.b);     //CE: b is not public in p1.A
                //System.out.println("c: "+a.c);     //CE: c has protected access in p1.A
                System.out.println("d: "+a.d);

                C c1 = new C();
                //System.out.println("a: "+c1.a);
                //System.out.println("b: "+c1.b);
                System.out.println("c: "+c1.c);
                System.out.println("d: "+c1.d);
        }
}
```

Only *protected* and *public* members are accessible.

Compilation:
```
D:\Naresh IT\HariKrishna\06AM>>javac -d . C.java
```

Execution:
```
D:\Naresh IT\HariKrishna\06AM>>java p2.C
c: 30    d: 40
```

**Accessing above four members from another package from normal class.**

```
//D.java
package p2;
import p1.A;
 class D {
        public static void main(String[] args) {
                A a = new A();
                //System.out.println("a: "+a.a);     //CE:  a has private access in p1.A
                //System.out.println("b: "+a.b);     //CE:  b is not public in p1.A
                //System.out.println("c: "+a.c);     // CE:  c has protected access in p1.A
                System.out.println("d: "+a.d);
                C c1 = new C();
                //System.out.println("c: "+c1.c);    // CE:  c has protected access in p1.A
                System.out.println("d: "+c1.d);
        }
}
```

It is not subclass so only *public* members are accessible.

Compilation:
```
D:\Naresh IT\HariKrishna\06AM>>javac -d . D.java
```

Execution:
```
D:\Naresh IT\HariKrishna\06AM>>java p2.D
d: 40
```