Access specifiers

Specifier	class	subclass	package	world
private	X			
public	X	X	X	X
protected	X	X*	X	
package	X		X	

* If a class is both a subclass of and in the same package as the class with the protected member, then the class has access to the protected member.

private

- a class member declared to be private is accessible from within the class in which it is defined:

```
public class Circle {
  private radius;
  public Circle(int r) {
✔     radius = r;
  }
  public boolean equals(Object o) {
✔     return radius == ((Circle)o).radius;
  }
}
```

private

```
public class FilledCircle extends Circle {
  private Color fillColour;
  public FilledCircle(int r, Color c) {
      super(r);
      fillColour = c;
  }
  public boolean equals(Object o) {
X     return radius == ((FilledCircle)o).radius &&
         fillColour == ((FilledCircle)o).fillColour;
  }
  public boolean equals(Object o) {
✔     return super.equals(o) &&
         fillColour == ((FilledCircle)o).fillColour;
  }
}
```

protected

- members declared protected can be used from within the same package.

```
package Greek;
public class Alpha {
  protected int p;
  protected void pMethod()
  {…}
}
```

```
package Greek;
class Gamma {
  void accessMethod() {
    Alpha a = new Alpha();
    a.p = 10; // legal
    a.pMethod(); // legal
  }
}
```

```
package Latin;
import Greek.*;
class Delta extends Alpha {
  void m(Alpha a, Delta d) {
    a.p = 10;    // illegal
    d.p = 10;    // legal
    a.pMethod(); // illegal
    d.pMethod(); // legal
  }
}
```

package or default

- if no access modifier is given then the access is implicitly package.
- classes in the same package can access package/default members.

```
package Greek;

class Alpha {
  int p;
  void pMethod()
  {…}
}
```

```
package Greek;

class Beta {
  void accessMethod() {
    Alpha a = new Alpha();
    a.p = 10; // legal
    a.pMethod(); // legal
  }
}
```

package/default classes

- default (unspecified) classes can live in the same file as a public class

```
public class LinkedList {
}
class Node {
}
```

- two public classes cannot be put in the same file.