

Assignment 8 (Due: 17th April before noon)

(S9.2) Fill in the following table to show how the different inheritance access specifiers affect the ability of the derived class's member functions to access a base class's members.

Member Restriction	this	Subclass	Package	General
public				
protected				
default				
private				

(S9.4) Suppose classes U.java contains the following definitions. What does the program output?

```
class S {
    private int value;

    public S(int x) {
        value = x;
        System.out.println("Made an S: " + value);
    }
    public int getValue() {
        return value;
    }
}
class T {
    private S s;

    public T(int x) {
        s = new S(x);
    }
    protected S getValue() {
        return s;
    }
}
```

```

public class U extends T {
    private U(int y) {
        super(y);
    }
    private void print() {
        S s = getValue();
        System.out.println(s.getValue());
    }
    public static void main(String[] args) {
        U u = new U(1128);
        u.print();
    }
}

```

(9.6) Consider the following file C.java.

```

public class C {
    protected int m;
    private int n;

    public C() {
        this(0,0);
    }
    C(int n) {
        this(n, 0);
    }
    protected C(int a, int b) {
        m = a;
        n = b;
    }
    int look1() {
        return m;
    }
    int look2() {
        return n;
    }
}

```

Suppose class D is extended from C. Which members of C can be accessed by D?
 How would your answer change (if at all), if class D was defined in the same file as C?

(9.8) Suppose class F has the following definition

```
public class F {  
    private int value;  
  
    public F(int x) {  
        value = x;  
        System.out.println("Made F: " + value);  
    }  
}
```

What is the output of the following program?

```
public class G extends F {  
    private F value;  
  
    public G(int x, int y) {  
        super(x);  
        value = new F(y);  
    }  
    public static void main(String[] args){  
        G g = new G(11, 28);  
    }  
}
```

For (9.15), study the following three programs first: ColoredPoint.java, ThreeDimensionalPoint.java, Colored3DPoint.java

(9.15) Suppose the following method main() was added to class Colored3DPoint.

What would the program output be?

```
public static void main(String[] args) {  
    Colored3DPoint c = new Colored3DPoint();  
    Colored3DPoint d = new Colored3DPoint(1, 2, 3, Color.black);  
    Colored3DPoint e = (Colored3DPoint) d.clone();  
    System.out.println(c);  
    System.out.println(d);  
    System.out.println(e);  
    System.out.println(d.equals(c));  
    System.out.println(d.equals(e));  
    return;  
}
```

For the following two programming problems, study the following three programs first: Fish.java, FishDemo.java, FishCanvas.java

Please make sure that your .java files can be compiled, or marks will be deducted

Also, marks will be deducted if the programs have no indentation

(9.17) Extend class Polygon by defining a subclass Triangle that implements the Drawable interface for representing triangles embedded in coordinate space.

(The following question is a bonus part)

For (9.20), also study the following three programs first: Box.java, Circle.java, GeometricObject.java

(9.20) Use instances of classes Polygon, Circle, and Box along with Triangle from Exercise (9.17) to paint a canvas with the following image.

