CSSE 220 – Object-Oriented Software Development
Rose-Hulman Institute of Technology

Worksheet 17

Name (Print):_____ Section:_____

1. **Access Modifiers (Review)**. Place access modifiers from most restrictive to least restrictive: _____

2. **Identify Access Modifiers**

   (a) _____: available to any other class in the same package

   (b) _____: can only be accessed within the declared class itself

   (c) _____: can be accessed by subclasses or any class in the same package

   (d) _____: can be accessed from any other class (everywhere)

3. **Scope and Lifetime (Variables)**:

   (a) _____: accessible anywhere inside the object

   (b) _____: accessible anywhere in the class (shared by all objects)

   (c) _____: accessible only within a method

   (d) _____: accessible only within a block (e.g., loop or if statement)

4. **Polymorphism Basics**. Consider the following code:

```
1 Enemy e1 = new Zombie();
2 e1.attack();
```

   (a) The **reference type** of `e1` is: _____

   (b) The **object type** created at run-time is: _____

   (c) Which type determines whether `attack()` is a **legal method call**? _____

   (d) Which type determines **which version of `attack()` runs**? _____

5. **Compile-Time vs Run-Time**

   (a) _____: checks whether a method call is allowed

   (b) _____: decides which overridden method executes

   (c) _____: based on the reference type

   (d) _____: based on the actual object

6.

```
interface Top {
    public void alpha();
    public void beta();
    public void gamma();
    public void delta();
    // Note no epsilon here
}
```

```
class One implements Top {

    public void alpha() {
        System.out.println("A");
    }

    public void beta() {
        System.out.println("B");
    }

    public void gamma() {
        System.out.println("C");
    }

    public void delta() {
        System.out.println("D");
        this.beta();
    }
}
```

```
class Two extends One
        implements Top {

    public void beta() {
        System.out.println("E");
    }

    public void gamma() {
        super.gamma();
        System.out.println("F");
    }

    public void epsilon() {
        System.out.println("G");
    }
}
```

Two m = new Two();
Top q = new One();
Top r = new Two();
One s = new Two();

Code	Output Choices (circle one in each problem)																
															no output	*runtime error*	*compile error*
`m.alpha();`	A	B	C	D	E	F	G	BE	CF	DB	DE	EB	FC				
															no output	*runtime error*	*compile error*
`m.gamma();`	A	B	C	D	E	F	G	BE	CF	DB	DE	EB	FC				
															no output	*runtime error*	*compile error*
`m.omega();`	A	B	C	D	E	F	G	BE	CF	DB	DE	EB	FC				
															no output	*runtime error*	*compile error*
`q.alpha();`	A	B	C	D	E	F	G	BE	CF	DB	DE	EB	FC				
															no output	*runtime error*	*compile error*
`r.beta();`	A	B	C	D	E	F	G	BE	CF	DB	DE	EB	FC				
															no output	*runtime error*	*compile error*
`r.epsilon();`	A	B	C	D	E	F	G	BE	CF	DB	DE	EB	FC				
															no output	*runtime error*	*compile error*
`s.beta();`	A	B	C	D	E	F	G	BE	CF	DB	DE	EB	FC				