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

Worksheet 14

Name (Print):________________________________ Section:______________

1. List 4 types of access modifiers in Java:

   1) ____________

   2) ____________

   3) ____________

   4) ____________

2. Complete with the names of modifiers

Modifier Name	Visibility
	Within the same class only
	Within same package only
	Same package + subclasses in any package
	Everywhere

3. Scenario 1: Player Inventory Questions: Which fields should be private, protected, or package-private? Which methods should be public vs. private?

   You are building an RPG. In your Player class you have:

   - ____________ String playerName
   - ____________ int level
   - ____________ List$< Item >$ inventory
   - ____________ void addItem(Item i)
   - ____________ void listInventory()
   - ____________ void applyInventoryDiscount() (helper method used internally)

   Additional Information from the software requirement specification:

   1) playerName and level track core state—hide them unless needed by subclasses

   2) addItem and listInventory form your public API

   3) applyInventoryDiscount is an implementation detail

4. Scenario 2: Game Engine Core

   Inside a gameengine package, you have:

   - Class GameObject with int x, y; and void render(Graphics g)
   - Class Engine in the same package needs to call render() but nothing outside gameengine should
   - Class SpriteObject in gameobjects package extends GameObject and overrides render()

   Questions:

   1) What modifier should GameObject's render have: public, protected, or package-private (=default)? ________________

   2) Should x, y be protected or private with getters?________________

   3) Should getters be private, public, protected? ________________

5. Scenario 3: Utility Helpers
   You create a helper class MathUtils in your game for math routines:

   - static int clamp(int value, int min, int max)
   - static boolean isPowerOfTwo(int n)

   Note: These methods are used across your entire application.

   Questions:

   1) Should MathUtils itself be public or package-private (default)? ________________

   2) What modifiers should be used for each method? ________________

6. Complete with the types of variables

Variable Name	Scope
	Anywhere in the class/object
	Throughout the method
	From declaration to the end of a block/method

7. Select the correct statements:
   1) Abstract Class cannot be instantiated on its own

   2) Abstract Class can be instantiated on its own like any Class

   3) Abstract class can provide both abstract methods (no implementation) and concrete methods (with implementation)

   4) Abstract class can only provide both abstract methods (no implementation)

   5) Abstract class can have constructors while Interface cannot have constructors

   6) Abstract class cannot have constructors similar to Interface

8. Whar are 4 steps to determine the method at Run Time?

   1) ________________

   2) ________________

   3) ________________

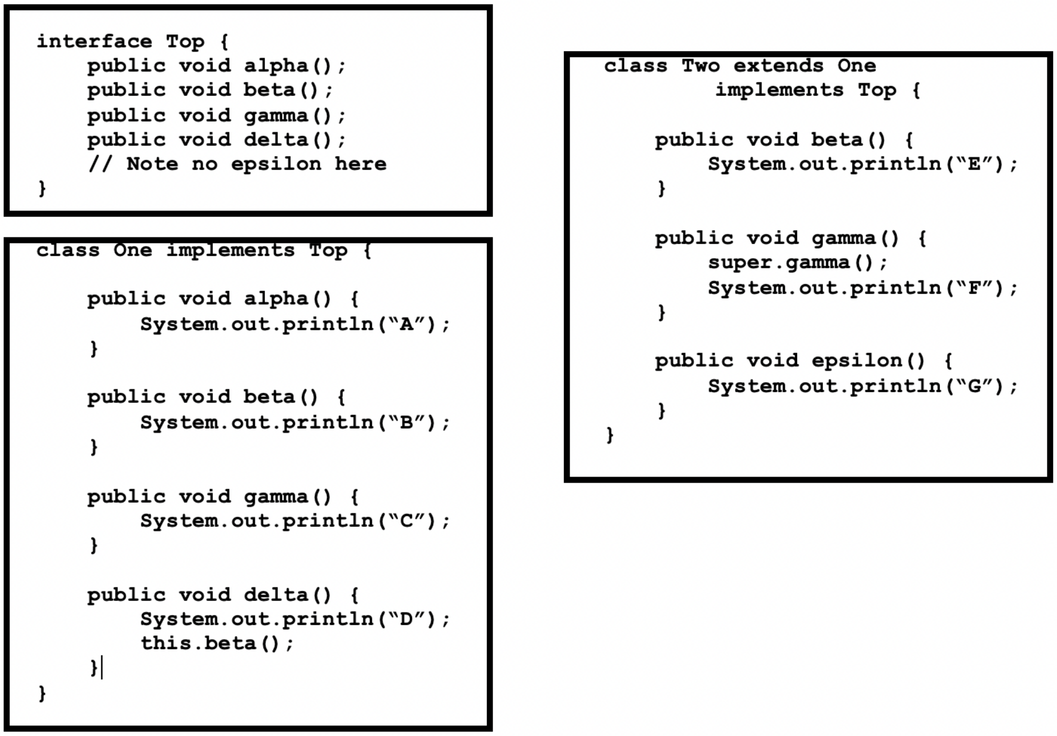
   4) ________________

9. What are common copile errors. Name at least three:

   1) ________________

   2) ________________

   3) ________________



```
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");
    }
}
```

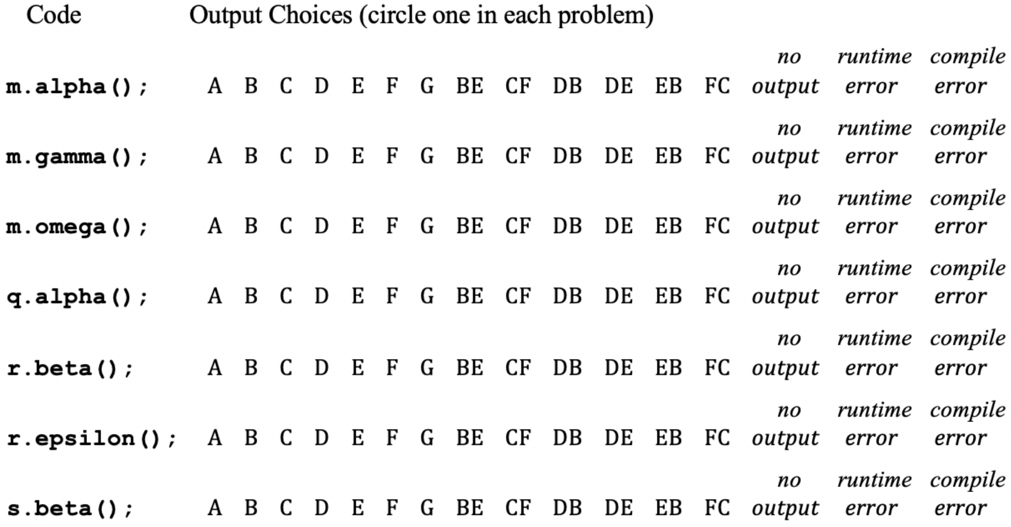
10\.

Suppose we declare and initialize these variables (all of these assignments are legal):
Two m = new Two();

Top q = new One();

Top r = new Two();

One s = new Two();



Code	Output Choices (circle one in each problem)															
`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	*no output*	*runtime error*	*compile error*