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

Worksheet 09

Name (Print):_____ Section:_____

1. **Review: When do we use `this`?**

   Circle all that apply.

   A) To refer to the current object's fields
   B) To distinguish instance variables from parameters
   C) To create a new object

2. **Will this compile?**

```java
public class Dog {
    private int age;

    public static void printAge() {
        System.out.println(this.age);
    }
}
```

   Reason: _____

3. **Classes vs Interfaces**

   Circle the correct statement:
   (A) Classes are blueprints and interfaces are contracts
   (B) Classes are contracts and interfaces are blueprints

4. **True / False**

   An interface specifies **what** a class can do without saying **how** it does it. _____

5. **Complete the code (syntax)**

   Fill in the missing keywords / headers.

```java
public ________ Animal {
    void animalSound();
}

class Dog ________ Animal {
    ________ void animalSound() {
        System.out.println("RRR");
    }
}
```

6. **Class or Interface?**

   Write **Class** or **Interface**.
   1) can create objects with `new` _____
   2) cannot be instantiated directly _____
   3) contains method implementations _____
   4) lists required method signatures _____

7. **Rules (memorize these)**

   Choose the correct word from the options in parentheses.

   A class _____ an interface (`extends` / `implements`)

   An interface can _____ another interface (`extends` / `implements`)

   A class can implement _____ interfaces (`one` / `many`)

   A class can extend _____ class (`one` / `many`)

8. **Implements vs Overrides**

   A class _____ an interface to agree to its contract (`implements` / `overrides`).

   A class _____ methods to provide the actual behavior (`implements` / `overrides`).

9. **Interface extends interface**

   You have two interfaces `Animal` and `Person`.
   You want `Person` to be a *special kind of* `Animal`.
   Complete the header:

```
1 public interface Person ________ Animal {
2     void talk();
3 }
```

10. **Reference type = what methods you can call**. Consider the following code

```
1 Person p = new Dog();
2 Dog d = new Dog();
```

    1) _____ provides access to all methods in class `Dog`.
    2) _____ only provides access to methods defined in `Person`.

    One-sentence why: _____

11. **Before You Leave**

    Write one question you still have about today's topic.