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

Worksheet 13

Name (Print):________________________________ Section:________________

1. Type of Programming:

   ________________: You call methods in sequence

   ________________: Your code reacts to events when they occur

2. Name the three main parts of the event model:

   1) ________________

   2) ________________

   3) ________________

3. Explain in one sentence: How does a custom ActionListener promote better encapsulation?

4. Select from the following: IS-A, HAS-A, DEPENDS-ON

   1) Dog implements Animal: ____________

   2) Class Zoo has a field - List$<$ *Animal* $>$: ____________

   3) Class Car passes a method drive() from the class Drivable into its method: ____________

   4) Dog extends Animal: ____________

5. Inheritance is a mechanism when a ____________ automatically gains the fields and methods of its ____________ by using the extends keyword

6. What is the name of the top most class in Java: ____________

7. Select the correct statements:

   1) A subclass inherits all fields and methods from its superclass, but cannot directly access private members

   2) A subclass does not inherit constructors from its superclass

   3) A subclass must invoke a superclass constructor (explicitly or implicitly) when its own constructor runs

   4) A subclass can override non-static, non-final methods that are public or protected

   5) A subclass cannot override private, static, or final methods from the superclass.

8. Invoke superclass:

   1) You have a superclass method drive(): ________________

   2) You have a superclass constructor Animal(String name): ________________

9. Write the output for each method below:

```java
class Animal {
    void move() { System.out.println("Animal is moving."); }
    void eat() { System.out.println("Animal is eating."); }
}

class Dog extends Animal {
    @Override
    void move()
    { System.out.println("Dog is running."); }
    void bark() { System.out.println("Dog is barking."); }
}

public class Geeks {
    public static void main(String[] args)
    {
        Dog d = new Dog();
        d.move(); // Output 1 : __________________
        d.eat(); // Output 2 : ____________________
        d.bark(); // Output 3 : ___________________
    }
}
```