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

Review Practice Exam 2

Name (Print):________________________ Section:____________

1. Time complexity is a measure of
   A) the total clock time taken by a program
   B) how many times a particular instruction set executes
   C) how much memory the program allocates
   D) how many input values the program can process

2. Selection Sort's worst-case running time grows as
   A) O(n)
   B) O(log n)
   C) O(n log n)
   D) O(n^2)

3. For the expression n/2 + n, the Big-O classification is
   A) O(1)
   B) O(log n)
   C) **O(n)**
   D) $O(n^2)$

4. **T** / F : Binary search requires the input array to be sorted

5. **T** / F : Selection Sort always makes the same number of comparisons, regardless of input order.

6. Polymorphic tracing

```
interface Animal {
    void speak();
    void move();
}
class Mammal implements Animal {
    public void speak() { System.out.print("growl "); }
    public void move()  { System.out.print("walk "); }
    public void feedYoung() { System.out.print("nurse "); }
}
class Dog extends Mammal {
    public void speak() { System.out.print("woof "); }
    public void move()  { super.move(); System.out.print("run "); }
    public void fetch() { System.out.print("fetch "); }
}
class Cat extends Mammal {
    public void speak()   { System.out.print("meow "); }
    public void scratch() { System.out.print("scratch "); }
}
```

```
Animal a = new Mammal();
Mammal  m = new Dog();
Animal  b = new Dog();
Mammal  c = new Cat();
Dog     d = new Dog();
Mammal  e = new Mammal();

```

#	Expression	Output / Error?
1	`a.speak();`	growl
2	`m.speak();`	woof
3	`b.move();`	walk run
4	`c.scratch();`	compile error
5	`((Cat) b).scratch();`	runtime error
6	`((Dog) b).fetch();`	fetch
7	`d.feedYoung();`	nurse
8	`((Animal) d).fetch();`	compile error
9	`((Dog) c).fetch();`	runtime error
10	`Animal x = new Animal();`	compile error

Note: Polymorphic traces: Method lookup uses the object's actual class (right side), not the declared type (left-side). Remember super.method() still invokes the superclass's code in relationship to where it is mentioned but this.method() invokes the actual class.

7. Which access level makes a class member accessible only within its own package?
A) public B) protected C) private **D) no keyword**

Which modifier allows a method to be overridden only by subclasses (in any package)?
A) public **B) protected** C) no keyword D) private

Which modifier makes a field constant at compile time?
A) static **B) final** C) abstract D) private

8. Identify the relationship: IS-A, HAS-A, DEPENDS-ON)

a. class Car extends Vehicle **IS-A**

b. class Library { List<Book>books; } **HAS-A**

c. class Pilot { void fly(FlightSimulator sim) { … }} **DEPENDS-ON**

d. class Cat implements Pet **IS-A**

e. class Smartphone { void charge(Battery b) { … }} **DEPENDS-ON**

9. Draw recursion traces for sumTo(4) and identify the final result (see worksheet 11 for proper drawing boxes)

```
 /**
 * Returns 1 + 2 +      + n
 */
public int sumTo(int n) {
    if (n <= 0) {
        return 0;
    }
    return n + sumTo(n - 1);
}

```

```
 sumTo(3)
    input = 3
    n <= 0? No
        return 3 + sumTo(2)

    sumTo(2)
      input = 2
      n <= 0? No
          return 2 + sumTo(1)

      sumTo(1)
        input = 1
        n <= 0? No
            return 1 + sumTo(0)

        sumTo(0)
          input = 0
          n <= 0? Yes
              return 0


```

then substitute blanks:

```
sumTo(1) returns 1 + [0]   = 1
sumTo(2) returns 2 + [1]   = 3
sumTo(3) returns 3 + [3]   = 6
sumTo(4) returns 4 + [6]   = 10

```

Final result: sumTo(4) returns 10