UNIVERSITY OF TWENTE.

PROGRAMMING

DIAGNOSTIC TEST 2: PUBQUIZ

ANSWERS ROUND 2

07 DECEMBER 2014
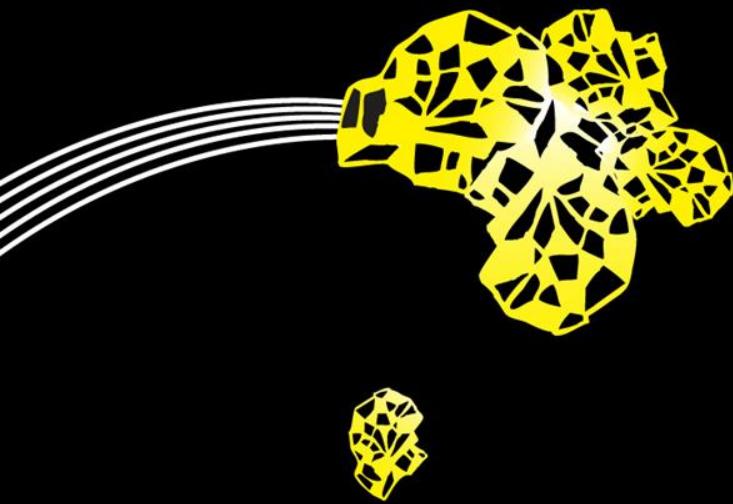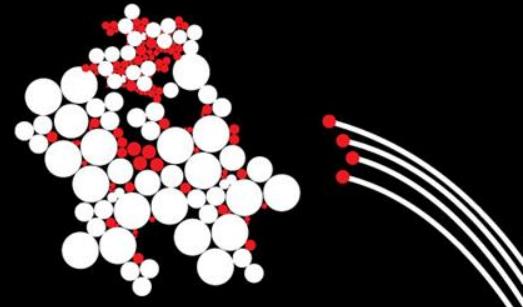
```java
class Rectangle {
    private int a;
    private int b;
    public void setA(int newA) {
        this.a = newA;
    }

    public void setB(int newB) {
        this.b = newB;
    }

    public int getArea() {
        return a * b;
    }
}

class Square extends Rectangle {
    public void setA(int a) {
        super.setA(a); super.setB(a);
    }

    public void setB(int b) {
        super.setB(b); super.setA(b);
    }
}
```

```java
class Test {
    public static void main(String[] args) {
        test(new Rectangle());
        test(new Square());
    }
    private static void test(Rectangle r) {
        r.setA(3);
        r.setB(4);
        System.out.println("Area: " + r.getArea());
    }
}
```

What does this print?

```java
class Point {
    boolean equal(Point x) { return false; }
}

class ColorPoint extends Point {
    boolean equal(ColorPoint x) { return true; }
}

class Puzzle {
    public static void main(String[] args) {
        Point p1 = new Point();
        Point p2 = new ColorPoint();
        System.out.println("1: " + p1.equal(p1));
        System.out.println("2: " + p2.equal(p1));

        ColorPoint cp = new ColorPoint();
        System.out.println("3: " + cp.equal(p2));
        System.out.println("4: " + p2.equal(cp));
        System.out.println("5: " + cp.equal(cp));
    }
}
```

What does this print?

```java
class A {
    int i = 1;
    int m() { return i * 100; }
    int n() { return i + m(); }
}
class B extends A {
    int i = 10;
    int m() { return i * 1000; }
}
class Test {
    public static void main(String[] args) {
        A x = new A();
        A y = new B();
        System.out.println("1: " + (x.i + y.m()));
        System.out.println("2: " + (y.i + x.m()));
    }
}
```

What does this print?

```
1.    interface A {
2.        abstract void gimme();
3.    }
4.
5.    class B extends A {
6.        public static final int value = 0;
7.
8.        int gimme() {
9.            value = value + 1;
10.           return value;
11.       }
12.
13.       abstract int B() {
14.           return gimme() > 0;
15.       }
16.   }
```

Which lines contain compiler errors?

```java
class A {
    int m() { return 4; }
    int n() { return this.m(); }
}

class B extends A {
    int m() { return 5; }
}
```

What does this print?

```java
class C extends B {
    int m() { return 3; }
    int n() { return super.m(); }
}
```

```java
public static void main (String [] args) {
    A v = new C();
    System.out.println("1: " + v.n());

    A w = new A();
    System.out.println("2: " + w.n());

    A x = new B();
    System.out.println("3: " + x.n());

    B y = new C();
    System.out.println("4: " + y.n());

    C z = new C();
    System.out.println("5: " + z.n());
}
}
```

```java
public class A {
    int i = 10;
    public void method(double i) {
        method((int) i);
    }
    public void method(int j) {
        System.out.println(j);
    }
    public void method() {
        method(i);
    }
    public static void main(String[] args) {
        A a = new A();
        a.method(5.0);
        a.method();
    }
}
```

What does
this print?

What is wrong with the following JML specifications?

```
class Rectangle {
    /*@
        requires a >= 0 & b >= 0;
        ensures \result == a * b;
    */
    public int getArea(int a, int b) {
        return a * b;
    }
}
class Square extends Rectangle {
    /*@
        requires a >= 0 & b >= 0 & a == b;
        ensures \result == a * a;
    */
    public int getArea(int a, int b) {
        return a * a;
    }
}
```

```
class Llama {
    public String color = "brown";
    public void sayHello() {
        System.out.println("I spit");
    }
}

class SuperLlama extends Llama {
    public String color = "black";
    public void sayHello() {
        System.out.println("I spit in your face");
    }
    public static void main(String[] args) {
        SuperLlama lama1 = new SuperLlama();
        lama1.sayHello();
        System.out.println(lama1.color);
        Llama lama2 = lama1;
        lama2.sayHello();
        System.out.println(lama2.color);
        System.out.println(lama2 instanceof SuperLlama);
    }
}
```

What does this print?

```
class A {
    public int getValue() {
        return 10;
    }
}
```

What is the type of the following expressions?

1. **new** A[] { }

2. **new** A().getValue()

3. ("Type of " + 1).length()

4. (**new** A[10])[3]

5. 4 == 1.0

```java
public class A {
    public static void main(String[] args) {
        int[] array = new int[] {0, 1, 2, 3, 4};
        for (int k = 0; k < array.length; k++) {
            int temp = array[k];
            array[k] = array[(k + 1) % 5];
            array[(k + 1) % 5] = temp;
        }
        for (int k = 0; k < array.length; k++) {
            System.out.println("" + (k+1) + ": " + array[k]);
        }
    }
}
```

What does this print?